

Sommersemester 2016

Java Programmierung im Bereich Internet of Things – Verwendung eines Raspberry Pi zur Erstellung von digitalen Einkaufslisten – Kann mit Hilfe eines Raspberry Pi das Einkaufen revolutioniert werden?

Thesis

zur Erlangung des Grades

Bachelor of Science

im Studiengang Wirtschaftsinformatik
an der Fakultät Wirtschaftsinformatik
der Hochschule Furtwangen University

vorgelegt von

Eduard Franke

Referenten:

Erstkorrektor: Professor Dr. Andreas Hess

Zweitkorrektor: Professor J. Anton Illik

Eingereicht am

5. August 2016

Eidesstattliche Erklärung

Ich, Eduard Franke erkläre hiermit an Eides statt, dass ich die vorliegende Bachelorthesis selbständig und ohne unzulässige fremde Hilfe angefertigt habe.

Die verwendeten Quellen sind vollständig zitiert.

Furtwangen, den 5. August 2016

Abstract

Der moderne Kühlschrank von heute hat viele Funktionen, die einige Jahre zuvor unvorstellbar waren. Die modernen Kühlschränke können sich selbst regulieren, sich selbst reinigen und können die Kühlluft gleichmäßig verteilen. Eine Innovation entsprang vom Unternehmen Samsung und seinem Modell T9000. Ein Kühlschrank, der mit dem Internet verbunden wird und somit das Leben von früher auf den Kopf stellt. Es besitzt ein Display mit dem alle Geräte im Haushalt gesteuert werden können. Die Voraussetzung ist jedoch, alle Geräte müssen mit dem Internet verbunden und auch steuerbar sein. Was der moderne Kühlschrank bis heute jedoch nicht kann, ist die Unterstützung der Menschen beim Einkaufen.

Das Schreiben von Einkaufslisten und der ständige Überblick was im Kühlschrank oder Speiseschrank verblieben ist und welche Produkte bereits konsumiert wurden, fällt vielen schwer oder ist einfach nur lästig. Beim Einkaufen werden benötigte Produkte vergessen und jene Produkte, die nicht benötigt werden, werden eingekauft. Das Führen von Einkaufslisten ist nervenzerreißend, da die Listen bis zum Einkauf gefüllt sind, jedoch beim Einkaufen entweder Zuhause oder im Auto vergessen werden. Dabei würde eine automatische Listenführung das Leben von vielen Menschen erleichtern. Die Einkaufsliste muss nicht mehr geführt werden. Sie aktualisiert sich selbst, wenn ein Produkt nicht mehr vorhanden ist und wird von dem Einkäufer aktualisiert, wenn das fehlende Produkt eingekauft worden ist.

Die vorliegende Thesis behandelt die Thematik „Java Programmierung im Bereich Internet of Things - Verwendung eines Raspberry Pi zur Erstellung von digitalen Einkaufslisten - Kann mit Hilfe eines Raspberry Pi das Einkaufen revolutioniert werden?“. Dabei wird in Kombination eines Raspberry Pi und einem Barcode-Scanner ein System entwickelt, das digitale Einkaufslisten erstellt und die Pflege der Produkte über das Internet ermöglicht.

Das Ziel dieser Bachelorthesis ist es, ein System zu entwickeln, das den Benutzer beim Einkaufen unterstützt und das Kaufen von falschen oder unnötigen Produkten vermeidet. Dabei wird aus der Kombination eines Raspberry Pi und einem Barcode-Scanner eine Produktverwaltungskomponente erstellt, die die verbrauchten Produkte einscannt und dabei automatisch eine Einkaufsliste erstellt, die die benötigten Produkte beinhaltet.

Inhaltsverzeichnis

1	Einleitung	19
1.1	Motivation	19
1.2	Projektbeschreibung.....	20
1.3	Aufbau der Thesis	21
2	Aufgabenstellung.....	23
2.1	Zielsetzung	23
2.1.1	Primäres Ziel	23
2.1.2	Sekundäre Ziele	24
2.1.3	Mögliche Problemfelder	24
2.2	Ähnliche Projekte	25
2.2.1	Samsung T9000	25
2.2.2	Family Hub Refrigerator.....	26
2.2.3	Amazon Dash.....	27
2.2.4	ZenFridge.....	29
2.2.5	Der intelligente Kühlschrank 2.0.....	30
2.3	Fachliches Umfeld.....	31
2.3.1	Internet of Things.....	31
2.3.2	Programmierungsumgebung	34
2.3.3	Hardware.....	39
2.4	Phasenplanung und Meilensteine des Projektes.....	46
3	Theoretischer Bezugsrahmen	47

3.1	Anforderungsdefinition	47
3.1.1	Systemkontext	47
3.1.2	Kurzbeschreibung der Akteure	48
3.1.3	Kurzbeschreibung der Use Cases	49
3.1.4	Anwendungsfalldiagramm	52
3.1.5	Fachklassenmodell	53
3.1.6	Systemfunktionalität	58
3.1.7	Beziehungen zwischen Anwendungsfällen	88
3.1.8	Nichtfunktionale Anforderungen	89
3.2	Systementwurf	91
3.2.1	Subsystemdiagramm	91
3.2.2	Verteilungsdiagramm	94
3.2.3	Benutzeroberfläche Webserver	95
3.2.4	Benutzeroberfläche JaRasPi	98
4	Praktischer Bezugsrahmen	99
4.1	Realisierung	99
4.1.1	Java-Anwendung	99
4.1.2	Website	110
4.2	Probleme	112
4.2.1	Artikelbezeichnung	112
4.2.2	Dateninkonsistenzen	113
4.2.3	Austragen aus der Produktliste	113
4.2.4	Datendarstellung auf der Webseite	113
4.3	Ausblick	114
4.3.1	Darstellung im Raspberry Pi anpassen	114
4.3.2	Subtrahieren	114
4.3.3	Rezepte anpassen/ändern	114
5	Fazit und Aussichten	115
5.1	Zusammenfassung	115
5.2	Schlussfolgerungen	116

Literaturverzeichnis	118
Anhang.....	124

Abbildungsverzeichnis

Abbildung 1: Amazon Dash-Fernbedienung	27
Abbildung 2: Amazon Dash-Button	28
Abbildung 3: ZenFridge.....	29
Abbildung 4: Der intelligente Kühlschrank 2.0	30
Abbildung 5: Das Java-Logo	34
Abbildung 6: HTML5 - Logo	36
Abbildung 7: Raspberry Pi 3 Modell B Nr.1.....	40
Abbildung 8: Raspberry Pi 3 Modell B Nr.2.....	40
Abbildung 9: USB-Netzteil Nr.1	41
Abbildung 10: USB-Netzteil Nr.2	41
Abbildung 11: HDMI-Kabel.....	42
Abbildung 12: SD-Karte mit 32 GB.....	42
Abbildung 13: Barcode-Scanner.....	43
Abbildung 14: Raspberry Pi Touch Screen mit 2x13 Pin Header-Verlängerung.....	44
Abbildung 15: Kamera Modul für Raspberry Pi mit Kameragehäuse Nr.1	44
Abbildung 16: Kamera Modul für Raspberry Pi mit Kameragehäuse Nr.2	45
Abbildung 17: Gesamtaufbau des Systems	45
Abbildung 18: Systemkontext der JaRasPi-Anwendung	48
Abbildung 19: Systemgrenzen	52
Abbildung 20: Klassendiagramm der JaRasPi-Anwendung	53
Abbildung 21: Klassenmodell	54
Abbildung 22: Systemablaufmodell Barcode scannen Nr.1	60
Abbildung 23: Systemablaufmodell Barcode scannen Nr.2.....	61
Abbildung 24: Systemablaufmodell Weiteres Produkt hinzufügen Nr.1	63
Abbildung 25: Systemablaufmodell Weiteres Produkt hinzufügen Nr.2	64
Abbildung 26: Systemablaufmodell Rezeptauswahl kontrollieren.....	66
Abbildung 27: Systemablaufmodell Einkaufsliste erstellen – Produktliste.....	69

Abbildung 28: Systemablaufmodell Einkaufsliste erstellen – Rezeptproduktliste.....	70
Abbildung 29: Systemablaufmodell Einkaufsliste erstellen – Getränke	71
Abbildung 30: Systemablaufmodell Einkaufsliste aktualisieren	73
Abbildung 31: Systemablaufmodell Foto erstellen	75
Abbildung 32: Systemablaufmodell Einkaufsliste auswählen	77
Abbildung 33: Systemablaufmodell Inhalt anzeigen	79
Abbildung 34: Systemablaufmodell Rezept auswählen	81
Abbildung 35: Systemablaufmodell Rezept erstellen	83
Abbildung 36: Systemablaufmodell Produkt entfernen	85
Abbildung 37: Systemablaufmodell Produkt hinzufügen	87
Abbildung 38: Beziehungen zwischen den Anwendungsfällen	88
Abbildung 39: Subsystemdiagramm Nr.1 – JaRasPi-Webserver	91
Abbildung 40: Subsystemdiagramm Nr.2 – MySQL-Datenbank.....	92
Abbildung 41: Subsystemdiagramm Nr.3 - JaRasPi-Anwendung.....	93
Abbildung 42: Verteilungsdiagramm	94
Abbildung 43: Startseite der Webseite	95
Abbildung 44: Einkaufsliste auswählen	95
Abbildung 45: Inhalt anzeigen.....	96
Abbildung 46: Rezept auswählen.....	96
Abbildung 47: Rezept-Einkaufsliste anzeigen	97
Abbildung 48: JaRasPi Startoberfläche.....	98
Abbildung 49: JaRasPi Oberfläche nach dem Start.....	98
Abbildung 50: Multithreading der Klasse HauptKlasse	100
Abbildung 51: While-Schleifen-Kette.....	100
Abbildung 52: Produkt einscannen	101
Abbildung 53: Produkt ausscannen	101
Abbildung 54: Übersetzung des Barcodes	102
Abbildung 55: Übersetzung des Barcodes mit der Bezeichnung.....	102
Abbildung 56: Korrektur des Übersetzten Barcodes	102
Abbildung 57: Datenbankverbindung	103
Abbildung 58: Anzahl verringern	103
Abbildung 59: Anzahl erhöhen.....	104
Abbildung 60: Rezeptauswahl Nr.1	104
Abbildung 61: Rezeptauswahl Nr.2.....	105
Abbildung 62: Rezeptauswahl Nr.3.....	105
Abbildung 63: Rezeptauswahl Nr.4.....	106
Abbildung 64: Rezeptauswahl Nr.5.....	106
Abbildung 65: Rezeptauswahl Nr.6.....	106

Abbildung 66: Einkaufsliste Nr.1.....	107
Abbildung 67: Einkaufsliste Nr.2.....	107
Abbildung 68: Einkaufsliste Nr.3.....	108
Abbildung 69: Einkaufsliste aktualisieren Nr.1	108
Abbildung 70: Einkaufsliste aktualisieren Nr.2	109
Abbildung 71: Kamera Nr.1	109
Abbildung 72: Kamera Nr.2	110
Abbildung 73: Einkaufsliste anzeigen	110
Abbildung 74: Rezept auswählen	111
Abbildung 75: Rezept erstellen	111
Abbildung 76: Produkt entfernen	111
Abbildung 77: Produkt hinzufügen.....	112

Tabellenverzeichnis

Tabelle 1:Phasenplanung und Meilensteine	46
Tabelle 2:Kurzbeschreibung der Akteure.....	48
Tabelle 3: Kurzbeschreibung der Use Cases.....	51
Tabelle 4: Klassenbeschreibung/Attributbeschreibung.....	57
Tabelle 5: Use Case 1 Barcode scannen	60
Tabelle 6: Use Case 2 Weiteres Produkt hinzufügen.....	63
Tabelle 7: Use Case 3 Rezeptauswahl kontrollieren	65
Tabelle 8: Use Case 4 Einkaufsliste erstellen.....	68
Tabelle 9: Use Case 5 Einkaufsliste aktualisieren	72
Tabelle 10: Use Case 6 Foto erstellen	74
Tabelle 11: Use Case 7 Einkaufsliste auswählen	76
Tabelle 12: Use Case 8 Inhalt anzeigen	78
Tabelle 13: Use Case 9 Rezept auswählen	80
Tabelle 14: Use Case 10 Rezept erstellen	82
Tabelle 15: Use Case 11 Produkt entfernen	84
Tabelle 16: : Use Case 12 Produkt hinzufügen.....	86

1 Einleitung

1.1 Motivation

Vielen Menschen ist die lästige Alltagssituation bekannt, man geht Einkaufen und stellt fest, dass die Einkaufsliste fehlt. Nun müsste man sich daran erinnern, was alles auf der Einkaufsliste stand. Nach dem Einkaufen muss jedoch festgestellt werden, dass benötigte Produkte nicht gekauft wurden. Und so muss erneut eingekauft werden. Die Lösung wäre eine Einkaufsliste, die ständig aufrufbar ist und immer aktualisiert vorliegt. Wie in dem Film „The 6th Day“¹, der Kühlschrank, der den Schauspieler Arnold Schwarzenegger daran erinnert, die Milch zu kaufen. So soll auch der moderne Kühlschrank, die Menschen von heute daran erinnern, was gekauft werden muss.

Die Idee des intelligenten Kühlschranks geht auf die Idee des Smart Home, wo alle Geräte eines Haushaltes miteinander verbunden sind und miteinander kommunizieren können, zurück. Der Benutzer hat ständig die Kontrolle und kann die einzelnen Geräte über ein Smartphone oder ein anderes Endgerät bedienen. Darunter fällt auch der Kühlschrank. Der Benutzer kann den Kühlschrank regulieren, das Innere des Kühlschranks ansehen und wird von diesem informiert, ob ein Produkt bald abläuft oder nachbestellt werden muss.

Aus der Idee wird 1999 Wirklichkeit. Der schwedische Elektronikhersteller „Electrolux“ veröffentlicht den ersten intelligenten Kühlschrank „Screenfridge“, der jedoch aufgrund von geringer Nachfrage nie auf den Markt kommt.² Weitere Hersteller wie LG oder Siemens präsentieren jedes Jahr neue intelligente Kühlschränke. Aber auch ihre Geräte kommen entweder nicht auf den Markt oder die Verkaufszahlen sind so gering, dass der Verkauf eingestellt werden muss. Ein Grund hierfür ist unter anderem der hohe Preis, den der Kunde für einen Kühlschrank mit einem integrierten Tablet zahlen muss.

¹ (ekitchen)

² Vgl. (Christe, 2002)

Die Idee des Smart Home kommt aus dem Bereich des „Internet of Things“ deutsch „Internet der Dinge“. Das Ziel ist die Vernetzung von allen Geräten, aber auch den Menschen mit den Geräten. Während die Geräte untereinander kommunizieren, kann sich der Benutzer über das Internet mit Hilfe eines Smartphones mit den Geräten verbinden, um Daten auslesen und Einstellungen vorzunehmen, um zum Beispiel in einem Haus die Heizungstemperatur zu regeln. Die ablaufende Kommunikation findet über das Internet statt. Das zur Verfügung stehende Gerät, wie in diesem Falle der Raspberry Pi, ist das „Ding“, das das „Internet of Things“ ausmacht.

Der Raspberry Pi ist ein Minicomputer, der über diverse Anschlüsse, wie z.B. USB oder LAN, verfügt. Zu Beginn wurde der Raspberry Pi nur für die Aneignung von Programmier- und Hardwarekenntnissen vorgesehen. Heute jedoch wird er nicht nur für Unterhaltungszwecke, sondern auch für große Projekte eingesetzt. Der Raspberry Pi unterstützt viele Programmiersprachen, was seine Einsatzmöglichkeit in verschiedenen Projekten ermöglicht. Der kleine Minicomputer wird von den Entwicklern oft, aufgrund des geringen Anschaffungspreises, bevorzugt.

Deshalb stellt sich die Frage, ob aus der Kombination des preiswerten Raspberry Pi und der Idee des intelligenten Kühlschranks ein funktionsfähiges System realisierbar wäre und ob das daraus entstandene System dem Endanwender einen Vorteil bieten würde?

1.2 Projektbeschreibung

Das Ziel dieser Thesis ist die Erstellung eines funktionstüchtigen Systems, welches den Benutzer beim Einkaufen unterstützen soll. Hierzu wird ein Raspberry Pi mit einem Barcode-Scanner, als die Hardware, ein Java-Programm, als die Software und ein Webserver, auf dem eine Webseite laufen wird, verwendet. Die Grundlage für das Erstellen des Systems setzt sich aus zwei Teilen zusammen. Zum einen aus der Analyse und Design des zu erstellenden Systems. In diesem Teil werden die Akteure und deren Beziehungen zu einander und den Systemen erklärt. Außerdem werden Use Cases beschrieben und dazu ein Anwendungsfalldiagramm erstellt.

Der wichtigste Abschnitt ist jedoch die Beschreibung der Systemfunktionalität. Hier wird jeder einzelne Use Case ausführlich beschrieben. Dabei wird auch der Ablauf im System Schritt für Schritt erläutert. Nach jedem Use Case wird das entsprechende Systemablaufmodell dargestellt. Nachdem alle Use Cases beschrieben wurden, werden die Beziehungen zwischen den Anwendungsfällen erläutert. Durch die Darstellung kann erkannt werden, welche Anwendungsfälle voneinander abhängig und welche voneinander getrennt sind. Als letztes erfolgt der Systementwurf. Neben der Darstellung des Subsystem- und des Verteilungsdiagramms wird hier die geplante Benutzeroberfläche dargestellt.

Der zweite Teil stellt die praktische Umsetzung des endgültigen Systems dar. Nachdem das System entwickelt worden ist, werden die bei der Realisierung entstandene Probleme und Lösungen beschrieben und ausführlich erklärt. Wichtige, komplexe oder besonders interessante Systemteile im Programmcode werden beschrieben und dargestellt.

Der Punkt Realisierung wird in zwei Teile aufgeteilt. Im ersten Teil wird die JaRasPi-Anwendung beschrieben und die wichtigen Umsetzungen erläutert. Dabei wird eingegangen wie der Barcode übersetzt und wie die Einkaufsliste aufgebaut wird. Im zweiten Teil werden die wichtigsten Punkte bei der Umsetzung der Webseite erläutert. Hierbei ist unter anderem das Auslesen der Einkaufsliste mit Hilfe von PHP oder das Erstellen von eigenen Rezepten aufgeführt.

Nachdem die Realisierung abgeschlossen wurde, werden die nicht gelösten Probleme und darauf aufbauend, der Ausblick für die JaRasPi-Anwendung erläutert.

1.3 Aufbau der Thesis

Die Thesis ist in drei Kapitel untergliedert. Das erste Kapitel behandelt die Aufgabenstellung. Hier wird zuerst die Zielsetzung für die Thesis festgesetzt und beschrieben. Es wird zuerst das primäre Ziel, die Anfertigung eines Systems für das Erstellen von digitalen Einkaufslisten und Rezeptverwaltung erläutert, im Anschluss folgt das sekundäre Ziel. Dies beschreibt den Einbau einer Kamera zum Fotografieren einer Fruchtschale oder das Innere eines Küchenschrankes. Danach werden ähnliche Projekte erläutert, gefolgt vom fachlichen Umfeld. Dabei wird neben den verwendeten Programmiersprachen, die Hardware beschrieben und der Begriff Internet of Things erläutert.

Das zweite Kapitel, der theoretische Bezugsrahmen wird in zwei Unterkapitel aufgeteilt. Das erste behandelt die Anforderungsdefinition. Hier werden neben der Beschreibung des Systemkontextes, die Akteure und die Use Cases beschrieben. Die Beziehung der Use Cases und der Akteure wird in einem Anwendungsfalldiagramm deutlich gemacht. Als nächstes wird das Fachklassendiagramm dargestellt. Es verdeutlicht, welche Klassen existieren und in welchen Beziehungen sie zu einander stehen.

Nachdem das Fachklassenmodell dargestellt wurde, werden die bereits kurz beschriebenen Use Cases ausführlich erläutert. Zusätzlich stellt das Systemablaufmodell den jeweiligen Use Case graphisch dar. Im Anschluss darauf werden die Beziehungen zwischen den Use Cases dargestellt. In diesem Punkt werden die Abhängigkeiten zwischen den Anwendungsfällen untereinander dargestellt, aber auch die Unabhängigkeiten zwischen den anderen Use Cases verdeutlicht.

Der zweite Teil des zweiten Kapitels ist der Systementwurf. In dem Punkt werden die softwarespezifischen Eigenschaften erläutert. Zuerst wird das Subsystemdiagramm dargestellt, welches den Aufbau des Systems verdeutlicht. Als nächstes wird das Verteilungsdiagramm erläutert. Dabei wird dem Leser verdeutlicht, welche Komponenten auf welcher Hardware laufen, wie sie konfiguriert sind und wie die Kommunikationsbeziehungen zwischen einander stattfinden. Das zu erstellende System wird eine Java-Anwendung besitzen, von der aus das Programm gestartet wird. Nebenläufig wird ein Apache Webserver aufgebaut, von dem aus auf die Einkaufsliste zugegriffen werden kann. Um dem Leser ein Verständnis zu geben, wie der Aufbau der Benutzeroberfläche aussehen könnte, wird im

Systementwurf eine beispielhafte Benutzeroberfläche, sowohl für die JaRasPi-Anwendung, als auch für die Webseite erstellt.

Das dritte Kapitel behandelt den praktischen Bezugsrahmen. Es wird in drei Unterkapitel aufgeteilt. Das erste Unterkapitel erläutert die Realisierung. Es ist der krönende Abschluss der Arbeit, denn hier wird vor allem die Darstellung der wichtigen, komplexen und der besonders interessanten Systemteile im Programmcode hervorgebracht. Es wird erläutert wie bestimmte Programmabläufe realisiert und welche Quellen dazu verwendet wurden. Nachdem die Realisierung erläutert wurde, werden die vorhandenen Probleme angesprochen. Probleme, wie Dateninkonsistenz oder weitere programminterne Probleme werden ausführlich beschrieben. Das dritte Unterkapitel ist der Ausblick. Es baut auf das Kapitel Probleme auf. Die Probleme werden erneut aufgegriffen, sie sind nicht unlösbar, können jedoch durch weiteren Zeiteinsatz zumindest teilweise gelöst werden.

Nachdem das System erstellt und getestet wurde, wird die Bedienungsanleitung für den Endverbraucher erstellt. Sie wird vor allem Übersichtsdarstellungen, wie der Benutzer die Einkaufsliste aufrufen oder diese bearbeiten, aber auch wie man eigene Rezepte erstellen und sie auswählen kann, beinhalten. Es wird erklärt, wie das entwickelte System zu bedienen ist und wie der Benutzer die JaRasPi-Anwendung zu benutzen hat. Außerdem wird ein Kapitel mit der Bezeichnung „Bereitstellung der Arbeitsumgebung“ erstellt, in dem erläutert wird, wie der Raspberry Pi zu konfigurieren ist, damit auch zukünftige Entwickler ihre Arbeit dort ansetzen können, wo die Thesis aufgehört hat.

Beide Kapitel sowohl die „Bedienungsanleitung“, als auch „Bereitstellung der Arbeitsumgebung“ können im Anhang nachgeschlagen werden.

2 Aufgabenstellung

2.1 Zielsetzung

Das Ziel der Bachelorthesis ist die Erstellung eines lauffähigen Systems. Mit diesem System ist es möglich, Produkte, die verbraucht wurden zu scannen und anschließend eine digitale Einkaufsliste mit den benötigten Lebensmitteln zu erstellen.

2.1.1 Primäres Ziel

Aus der Kombination eines Raspberry Pi und einem Barcode-Scanner wird ein System mit dem Namen „Ja-RasPi“ erstellt, das folgende Aufgaben erledigen wird:

- Ist ein Produkt verbraucht, wird dessen Barcode zum Beispiel eine Milchpackung, mit dem Barcode-Scanner gescannt
- Wurde die Milchpackung gescannt, wird der Barcode mit Hilfe des Raspberry Pi unter Zuhilfenahme einer Webseite übersetzt
- Nach der Übersetzung wird das Produkt als das gespeichert, was es ist und zwar eine verbrauchte Packung Milch
- Die Speicherung erfolgt in einer Datenbank
- Auf Basis dieser Datenbank wird eine digitale Einkaufsliste aufgebaut
- Damit die Einkaufsliste auf allen Endgeräten angezeigt werden kann, wird ein Webserver erstellt, von dem aus auf eine Webseite zugegriffen werden kann, auf der die Einkaufsliste dargestellt wird

Das fertiggestellte System wird in der Küche angebracht und hat eine dauerhafte Verbindung mit dem Internet. Dabei beschränkt sich das System nicht nur auf den Kühlschrank, sondern wird universell für alle mit einem Barcode versehenen Produkte angewendet.

Die Funktion des Systems beschränkt sich nicht nur auf das Erstellen von Einkaufslisten. Das System soll außerdem Rezepte, die über die Webpräsenz aufgerufen werden, erkennen und die benötigten Produkte in die Datenbank aufnehmen. Das Programm kontrolliert, ob die benötigten Zutaten vorhanden sind oder nicht. Sind Produkte nicht vorhanden, werden diese in einer neuen Datenbank gespeichert. Der Benutzer kann aus der Einkaufsliste sowohl die eingescannte Produkte, als auch die Zutaten für die ausgewählte Rezepte entnehmen.

Wurden Produkte gekauft, werden sie durch das Scannen mit dem Barcode-Scanner wieder in die Datenbank aufgenommen und als vorhanden verbucht.

2.1.2 Sekundäre Ziele

Neben dem primären Ziel, der Erstellung von digitalen Einkaufslisten, werden weitere Ziele genannt. Da die Arbeit mit dem Raspberry Pi viele Möglichkeiten zur Erweiterung des Systems offen lässt, sind während der Ideenfindungsphase weitere Optimierungsmöglichkeiten entstanden.

- Einbau und Programmierung einer Kamera über einer Fruchtschale oder in einem Lebensmittelschranks, um in bestimmten Abständen ein Foto von dem Inhalt der Fruchtschale oder eines Lebensmittelschranks zu schießen

Der Inhalt der Fruchtschale oder das Innere des Lebensmittelschranks kann wie die digitale Einkaufsliste auf der Webpräsenz aufgerufen werden. Wenn das primäre Ziel aus Kapitel 2.1.1 frühzeitig abgeschlossen wird, werden die sekundären Ziele abgearbeitet und dem Endsystem hinzugefügt.

2.1.3 Mögliche Problemfelder

Da die Bachelorthesis einen Programmierteil beinhaltet, muss mit Schwierigkeiten gerechnet werden. Bei der Ideenfindungsphase wurden einige Problemfelder definiert. Die unten aufgeführten Punkte sind die möglichen Problemfelder, die während der Bachelorthesis auftreten können.

- Die Kommunikation zwischen Raspberry Pi und dem Barcode-Scanner funktioniert nicht einwandfrei
- Beim Einscannen des Produktes besteht die Gefahr, dass der Barcode des Produktes mehrere Male gescannt wird, dadurch wird der Artikel öfter als gewollt in die Datenbank aufgenommen

- Der eingescannte Barcode muss mit Hilfe von Online-Datenbanken übersetzt und als der benötigte Artikel in die Datenbank eingetragen werden
- Bei der Erkennung von benötigten Zutaten aus einem ausgewählten Rezept, muss mit Schwierigkeiten gerechnet werden, da es unterschiedliche Bezeichnungen der Zutaten geben könnte

2.2 Ähnliche Projekte

Die Idee eines intelligenten Kühlschranks gibt es schon lange. Anfang des 21ten Jahrhunderts wurde von Electrolux ein innovativer Kühlschrank vorgestellt. Der Screenfridge war mit einem 15 Zoll Bildschirm ausgestattet, besaß Lautsprecher und eine Digitalkamera. Neben der Kühlfunktion konnte der Kühlschrank als Fernseher, Radio und Surf-Station genutzt werden. Weiterhin konnte der Screenfridge Kochrezepte verwalten, die Webcam konnte zum Telefonieren benutzt werden und der Inhalt des Kühlschranks konnte digital auf dem Bildschirm angezeigt werden. Eine weitere Innovation von Electrolux war die Selbstdiagnosefunktion. Mit dieser Funktion konnte der Benutzer darüber informiert werden, ob Produkte im Inneren des Kühlschranks abgelaufen sind.³

Damit aus der Idee Wirklichkeit werden kann, wurde im Jahre 2000 von Ericsson und Electrolux ein Gemeinschaftsprojekt, mit dem Namen e2Home, ins Leben gerufen. Das Ziel war es, in einem Feldversuch herauszufinden, ob sich ein Internet-Kühlschrank im alltäglichen Leben durchsetzt. Das Ergebnis war jedoch ernüchternd, da das System viele Fehler aufwies und zu den damaligen Zeiten noch nicht ausgereift war. Aus diesen Gründen hat sich der Screenfridge nicht durchgesetzt.⁴

2.2.1 Samsung T9000

13 Jahre später startet Samsung einen neuen Versuch. Auf der CES 2013 in Las Vegas stellt Samsung den T9000 vor. Neben der Kühlfunktion hat der Kühlschrank einen 10 Zoll Touchscreen, welcher mit Android betrieben wird. Auf dem integrierten Computer lassen sich zwar keine eigenen Applikationen installieren, da die vorinstallierten Apps für den Gebrauch völlig ausreichend sind.⁵ Neben dem Google Kalender und der Nutzung des Internets lassen sich über den Internetdienst Evernote, Einkaufslisten erstellen und auf andere Android Geräte verschicken. Eine

³ Vgl. (Christe, 2002)

⁴ Vgl. (Borchert, 2000)

⁵ Vgl. (Pasolini, 2013)

weitere Funktion, die der T9000 besitzt, ist das Erstellen von Rezeptvorschlägen oder das Mitteilen, wann ein bestimmtes Produkt abzulaufen droht. Diese Funktionen setzen jedoch das menschliche Handeln voraus. In der Zukunft sollte eine Barcodescan-Funktion eingebaut werden, was bis heute jedoch nicht geschehen ist.⁶

2.2.2 Family Hub Refrigerator

Drei Jahre später versucht Samsung es erneut und veröffentlicht Anfang 2016 den Family Hub Refrigerator. Neben der Kühlfunktion hat das Gerät einen 21,5 Zoll Display auf dem verschiedene Programme verwendet werden können. Wie bei dem T9000 lassen sich unter anderem auch hier ein Kalender, Rezepte und Einkaufslisten anzeigen. Eine Neuerung zu dem T9000 ist die Fernseh- und Musikstreaming-Funktion.

Eine Innovation sind die drei Kameras, die im Inneren des Kühlschranks angebracht wurden. In Verbindung mit einer Applikation lassen sich die Kameras aktivieren und der Inhalt des Family Hub Refrigerator kann angezeigt werden. Es kann aus beliebigen Orten auf die Kameras zugegriffen werden, sogar mit einem Smartphone beim Einkaufen. Das heißt, wenn man sich nicht sicher ist, ob ein Produkt vorhanden ist, lässt sich durch die Applikation der Inhalt des Kühlschranks sofort anzeigen. Das Problem des bequemen Einkaufens von Zuhause aus, hat Samsung auch mit dem Family Hub Refrigerator nicht realisiert. Um dieses ehrgeizige Projekt zu erreichen hat Samsung Mitarbeiter von Mastercard angeworben. Sie sollen eine Applikation mit dem Namen Groceries programmieren, welche einen Einkauf über den Family Hub's Touchscreen ermöglichen soll.⁷

Zusammenfassend sind jedoch alle Geräte, sowohl der Screenfridge von LG Electronics mit ca. 5000 Euro⁸, als auch Samsung's T9000 in Höhe von 3999 US \$⁹ oder der Family Hub Refrigerator ab der Summe von 5799 US \$¹⁰, eine teure Investition. Für einen einfachen Bürger bringt der Family Hub Refrigerator bisher keine wesentlichen Vorteile mit sich. Außerdem würde für viele ein Kühlschrank der genauso viel kostet wie ein Kleinwagen, nicht in Frage kommen. Für die Meisten ist ein Kühlschrank für das Kühlen zuständig. Die Nebenfunktionen, wie das Fernsehen oder die Sicht in das Innere des Kühlschranks sind zwar interessant und in vielen Situationen auch hilfreich, jedoch überwiegt der allzu hohe Preis die Vorteile der Nebenfunktionen.

⁶ Vgl. (Grün, 2014)

⁷ Vgl. (Lavars, 2016)

⁸ Vgl. (Christe, 2002)

⁹ Vgl. (Pasolini, 2013)

¹⁰ Vgl. (Samsung)

Diese Entwicklung konnte man sowohl bei dem T9000, als auch bei dem Screenfridge beobachten. Beide Geräte hatten innovative Technologien, wurden von den Bürgern jedoch nicht akzeptiert und wurden auch nie marktreif. Bei dem Screenfridge spielte sowohl der Preis, als auch die Tatsache, dass der Kühlschrank eher als ein „Spaßfaktor“ angesehen wurde, eine entscheidende Rolle. Aber selbst auch das bequeme Einkaufen, war für die Haushalte sehr wichtig, denn hätte der Screenfridge die Benutzer informieren können, ob die Produkte im Inneren schlecht geworden sind, wären sie auch bereit gewesen einen intelligenten Kühlschrank zu kaufen.¹¹

2.2.3 Amazon Dash

Das bequeme Einkaufen wird auch von Amazon angestrebt. Mit Amazon Dash einem Handscanner will der amerikanische Online-Versandhändler im Jahr 2014 das Einkaufen revolutionieren. Wie in der Abbildung 1 abgebildet ist der Handscanner nicht größer als eine Fernbedienung und wird zum Einscannen von Barcodes benutzt. Wurde ein Produkt eingescannt, wird dieses in das Amazon Fresh-Konto aufgenommen. Weiterhin kann der Benutzer mit dem eingebauten Mikrofon den Namen des Produktes sagen, um es in den Warenkorb aufzunehmen.¹²



Abbildung 1: Amazon Dash-Fernbedienung¹³

¹¹ Vgl. (Borchert, 2000)

¹² Vgl. (Dirscherl, 2014)

¹³ (Amazonfresh)

Die Voraussetzung für die Nutzung ist ein Amazon Fresh-Konto und ein Wohnort in Los Angeles, San Francisco oder Seattle, denn nur dort wird das Amazon Dash eingesetzt. Weiterhin muss die Bestellung nachdem die Produkte eingescannt oder mit der Stimme in den Warenkorb aufgenommen wurden, mit einem Smartphone oder Computer bestätigt werden.¹⁴

Ein Jahr nach der Einführung des Amazon Dash wird das Projekt aufgegeben und ein neues Projekt, Amazon Dash-Button gestartet. Mit Hilfe des Dash-Buttons können Produkte durch das einfache Klicken auf den Button bestellt werden. Nach dem Klicken muss eine Bestellbestätigung entweder mit dem Smartphone oder Computer erfolgen. Für bestimmte Produkte gibt es je einen Dash-Button, wie zum Beispiel in der Abbildung 2 für das Spülmaschinenprodukt Finish. Das Dash-Button-Angebot ist momentan nur in Amerika verfügbar und ist nur für ausgewählte Prime-Mitglieder nutzbar.¹⁵¹⁶



Abbildung 2: Amazon Dash-Button¹⁷

Mit dem Dash-Button kommt Amazon dem bequemen Einkaufen einen großen Schritt näher. Aber die Vorstellung mehrerer 100 Dash-Buttons in einem Lebensmittelschrank oder auf einem Kühlschrank, auf dem je nur ein Produkt abgebildet ist, ist nicht unbedingt eine bequemere Art einzukaufen. Vor allem dann nicht, wenn man bedenkt, dass nach jedem Dash-Buttonklick eine Bestellbestätigung erfolgen muss. Der Amazon Dash Handscanner war viel praktischer, da nur ein Gerät benötigt wurde und durch das Einscannen oder Aussprechen des jeweiligen Produktes, es zumindest theoretisch möglich war, alle Produkte zu bestellen.

¹⁴ Vgl. (Dirscherl, 2014)

¹⁵ Vgl. (Fuchs, 2015)

¹⁶ Vgl. (Horchert, 2015)

¹⁷ (Amazon)

Alle bisherigen Lösungen haben gute Ideen, kommen aber dem Benutzer nicht entgegen. Das Ziel, das Einkaufen zu revolutionieren oder zu verbessern wird nicht erfüllt. Entweder muss sich der Verbraucher einen teuren Kühlschrank mit einem eingebauten Computer, dessen Funktionen noch nicht ausgereift sind, kaufen. Oder sich 100te verschiedene Knöpfe kaufen und sie im Haus verteilen, um ein verbrauchtes Gut bequem kaufen zu können. Doch allein die Vorstellung von unzähligen Knöpfen am Kühlschrank oder in einem Lebensmittelschrank lässt die Idee des bequemen Einkaufens nicht so revolutionär erscheinen.¹⁸ Die nächsten zwei Kapitel handeln von privaten Projekten, die auf das Erstellen von Einkaufslisten und das Optimieren des Kühlschranks eingehen.

2.2.4 ZenFridge

Der ZenFridge ist nichts anderes als ein Kühlschrank, der mit zwei eingebauten Waagen als Ablageflächen und einem Barcode-Scanner erweitert wurde. Der Scanner liest, wie in der Abbildung 3 abgebildet, beim Reinlegen die Produkte. Bei der Entnahme werden die Produkte erneut eingescannt, um das Fehlen des Produktes zu vermerken. Wird das Produkt jedoch wieder zurückgelegt, muss es zuerst erneut eingescannt werden und anschließend auf die eingebaute Waage gelegt werden, damit das Gewicht berechnet werden kann. Somit kann die Restmenge ermittelt und bei Bedarf, zum Beispiel mit einem Smartphone, aufgerufen werden.



Abbildung 3: ZenFridge¹⁹

¹⁸ Vgl. (Horchert, 2015)

¹⁹ (Sonntag, 2012)

Eine weitere Funktion des ZenFridge, beziehungsweise der ZenFridge-Software, ist das Verwalten und Auslesen der Lieblingsrezepte. Aber auch das Erstellen von Einkaufslisten für die Produkte, die einem für das Lieblingsrezept noch fehlen. Weiterhin hat die ZenFridge-Software die Funktion, den Verbrauch der letzten Monate zu verwalten. Dabei können verschiedene Auswertungen vorgenommen werden, die zum Beispiel einem Menschen in einer Diätphase helfen können.²⁰

Das ZenFridge Projekt kommt dem Ziel, das Einkaufen zu revolutionieren und es bequemer zu gestalten, näher als Amazon, Samsung oder LG. Zum einen werden die Produkte über einen Barcode-Scanner in die Datenbank aufgenommen. Zum anderen können Einkaufslisten für ausgewählte Rezepte erstellt werden, was die zwei Hauptfunktionen des ZenFridge sind. Jedoch wird auch hier auf das Einkaufen selbst, nicht eingegangen. Bei dem nächsten Projekt wird auf die Problematik des bequemen Einkaufens mehr eingegangen.

2.2.5 Der intelligente Kühlschrank 2.0

Das Projekt „Der intelligente Kühlschrank 2.0“ wurde von einer Arbeitsgruppe an dem Ratsgymnasium in Osnabrück, im Rahmen einer Projektarbeit mit dem Titel „Jugend forscht“ durchgeführt. Wie in Abbildung 4 dargestellt, scannt der Benutzer mit einem Barcode-Scanner das Produkt ein. Die Daten werden in einem Raspberry Pi eingelesen und über eine Internetverbindung an einen FTP-Server weitergeleitet. Nachdem die Daten auf dem Server gespeichert wurden, können sie über ein Smartphone aufgerufen werden.

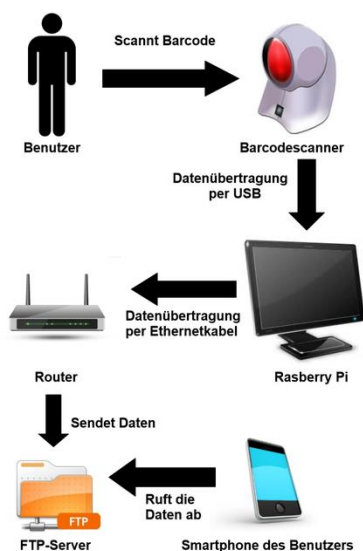


Abbildung 4: Der intelligente Kühlschrank 2.0²¹

²⁰ Vgl. (Sonntag, 2012)

²¹ (Kasten)

Zum Erstellen der benötigten Software wurde die Programmiersprache Python verwendet. Als Hardware wird ein Raspberry Pi Model B+ und zwei Barcode-Scanner verwendet.²²

Dieses Projekt ähnelt der Aufgabenstellung dieser Thesis, denn die Hardwareseite und zwar ein Raspberry Pi und ein Barcode-Scanner, werden ebenfalls zum Erstellen des Systems verwendet. Die Softwareseite wird sich jedoch unterscheiden. Es werden zwei Softwaresysteme erstellt, die auf dem Raspberry Pi parallel laufen werden und zwar eine Java-Anwendung und ein Apache Webserver. Das Ziel der Bachelorarbeit ist eine Kombination aus der Zielsetzung des ZenFridge und des intelligenten Kühlschranks 2.0, mit der Ergänzung, Kochrezepte zu verwalten und auf deren Basis Einkaufslisten zu erstellen.

2.3 Fachliches Umfeld

Das Kapitel 2.3 beschreibt das Umfeld in dem sich die Thesis befindet. Neben der Beschreibung der fachlichen Begriffe wird in diesem Kapitel auf die verwendete Hardware eingegangen. Es werden Unterschiede zwischen den Raspberries Pi Versionen gezogen und darauf eingegangen, weshalb eine bestimmte Version verwendet wurde. Weiterhin werden die Programmiersprachen erläutert, die für die Programmierung der Applikation und der Webseiten verwendet werden.

2.3.1 Internet of Things

Internet of Things, deutsch Internet der Dinge, ist heutzutage in aller Munde und bedeutet so viel wie die Verbindung der vielen Geräte miteinander, aber auch mit dem Menschen, der die Geräte verwendet. Die Experten der internationalen Organisation IEEE schätzen, dass das gewaltige Netz, welches sich im Moment ausbaut, in wenigen Jahren mehr als 50 Mrd. Dinge miteinander verbinden wird. Bis zum Jahr 2020 sollen es mehr als 100 Mrd. sein. Die Verbindung findet dabei nicht nur zwischen den Maschinen, Fahrzeugen und der Haustechnik statt, sondern auch mit dem Menschen. Dabei nutzt der Mensch ihm verfügbare Technologien, wie zum Beispiel das altbekannte Smartphone.²³

²² Vgl. (Kasten)

²³ Vgl. (Andelfinger, et al., 2015), S. 9.

2.3.1.1 Einleitung

„In der gegenwärtigen technologischen Entwicklung ist eine immer stärkere Integration von Sensoren und Prozessoren in Alltagsgegenstände zu beobachten. Ermöglicht durch die zunehmende Miniaturisierung der Komponenten, deren sinkende Preise und den geringer werdenden Energieverbrauch, können Informationsverarbeitung und Kommunikationsfähigkeit selbst in Gegenstände vordringen, die auf den ersten Blick keine elektronischen Geräte darstellen. Diese Gegenstände (Dinge) werden dadurch mit technischer Intelligenz ausgestattet, können (Umgebungs-) Informationen aufnehmen und teilweise autonom auf deren Basis handeln. Durch zunehmende Vernetzung und den Austausch von Informationen untereinander entsteht ein Internet der Dinge, in dem neben Menschen Gegenstände bzw. Systeme die eigentlichen Akteure sind.

2.3.1.2 Internet der Dinge vs. Internet der Menschen

Das Internet der Dinge ist ein Trend, der in letzter Zeit immer häufiger in den Medien Erwähnung findet. Doch worum handelt es sich dabei genau und worin unterscheidet es sich vom uns bisher bekannten Internet? Man kennt bisher das Internet vor allem von der Nutzung mit dem Computer. Der Mensch benutzt einen PC, um explizit seine gewünschten Informationen abzurufen. Auch durch Smartphones hat sich die bisherige Nutzungsweise des Internets nur wenig verändert. Lediglich die Größe des Endgeräts und der Grad der Mobilität unterscheiden sich. Der Faktor Mensch stand bisher bei der Nutzung des Internets im Vordergrund. Zukünftig könnte sich die Nutzung durch das Internet der Dinge jedoch grundlegend ändern(...). Der Trend geht zu Alltagsgeräten, die mit Sensoren und Internetanbindung ausgestattet werden. Mit einem klassischen PC haben diese Gegenstände nichts mehr gemeinsam (...) und doch hat man die Möglichkeit, das Internet zur Kommunikation zu nutzen. Eine weitere Eigenschaft dieser Geräte ist die Möglichkeit zur autonomen Kommunikation. Die Geräte sind nicht mehr auf Personen angewiesen, die sie bedienen, sondern können selbstständig (sogar untereinander) kommunizieren, ohne dazu angeleitet zu werden. Der Faktor Mensch gerät zunehmend in den Hintergrund, während die Gegenstände in den Vordergrund rücken (...).

2.3.1.3 Aktuelle Anwendungsgebiete

Das Internet der Dinge hält schleichend Einzug in unseren Alltag. Ermöglicht wird dies durch den Einsatz verschiedener Techniken (...). Die Grundlage hierzu bildet vor allem der stetige Fortschritt in den Bereichen Mikroelektronik, Kommunikationselektronik und Informationstechnologie. Die Miniaturisierung und gleichzeitige Leistungssteigerung in diesen Technologiebereichen lässt bisher ungeahnte Einsatzgebiete zu. Die Vision einer weltweiten Vernetzung von Gegenständen rückt damit immer näher. Bereits heute lassen sich viele einfache, aber auch komplexere Anwendungsbeispiele finden:

Beim „Intelligenten Wohnen“ werden Geräte in privaten Wohnhäusern miteinander vernetzt. Dies reicht von der Haustechnik (Licht, Alarmanlage, Heizung, Rollläden), über Verbrauchszähler („Smart Metering“), elektrische Haushaltsgeräte (Herd, Kühlschrank, Waschmaschine) bis hin zu Multimedia-Geräten (Fernseher, Radio, Computer) und dem Internetzugang. Dadurch kann die Heizung beispielsweise bedarfsgerecht und individuell abhängig von den anwesenden Bewohnern gesteuert werden.

2.3.1.4 Risiken

Das Internet der Dinge bietet ein erhebliches Potenzial; die Gesellschaft wird sich künftig sogenannten „smarten Dingen“ gegenübersehen. Die sich bietenden Chancen sind vor allem davon abhängig, wie die Menschen die neuen Möglichkeiten annehmen werden. Technischer Fortschritt wird im Allgemeinen positiv bewertet, jedoch kommt es gerade beim Internet der Dinge darauf an, dass der Nutzen wahrnehmbar ist (...). Der zukünftigen Anwendung des Internet der Dinge sind fast keine Grenzen gesetzt. Doch was bedeutet dies für unsere Gesellschaft? Zum einen bringt die Integration von Technik und Kommunikation in alltägliche Gegenstände je nach Verwendung eine große Erleichterung. Die Gegenstände kommunizieren miteinander und erfassen Daten. Der Mensch muss sich nicht mehr selbst um Dinge des Alltags kümmern, da die Gegenstände dies nun autonom anhand der interpretierten Daten aus ihren Sensoren erledigen oder zumindest eine Benachrichtigung an das Smartphone schicken. Als Beispiel sei der berühmte „internetfähige Kühlschrank“ genannt, der selbstständig Lebensmittel nachbestellt (...).

Jedoch haben diese Vorzüge auch ihre Schattenseiten. Der Mensch wird immer mehr abhängig von Technik und verlässt sich auf diese. Doch Technik kann ausfallen. Mit Blick auf das Beispiel „Internet-Kühlschrank“ würde dies bei Ausfall der Internetverbindung bedeuten, dass der Kühlschrank leer ist, weil die Lebensmittelbestellung nicht übermittelt werden konnte. Schlimmer wiegt das Beispiel Autoreifen. Verlässt sich der Fahrer zu sehr auf die Technik, statt gelegentlich selber den Druck des Reifens zu überprüfen, könnte ein defekter Reifen, der aufgrund eines Fehlers des Sensors nicht gemeldet wird, bei hohen Geschwindigkeiten zu ernsthaften Schäden führen. Kurz gesagt kann die Technisierung von alltäglichen Gegenständen unser Leben durchaus erleichtern; jedoch muss man sich immer bewusst sein, dass Technik ausfallen kann.

Ein ganz anderer Aspekt ist das Thema Datenschutz (...). Durch die Verwendung von Sensoren in alltäglichen Geräten wissen die Geräte immer mehr über ihren Besitzer und dessen (Alltags-) Leben. Was mit diesen Daten passiert bzw. welche anderen Geräte diese unbemerkt auslesen und wo die Daten zusammengeführt und weiterverarbeitet werden, ist wenig transparent und kontrollierbar. Es gilt hier sowohl ein Bewusstsein für den Datenschutz beim Anwender zu entwickeln, als auch Maßnahmen auf anderen Ebenen zu ergreifen, um dem Kontrollverlust nachhaltig entgegenzuwirken.²⁴

²⁴ (Mehler-Bicher, et al., 2012), S. 118 – 121.

2.3.2 Programmierumgebung

In dem nächsten Kapitel wird die Programmierumgebung erläutert, die für die Realisierung der Thesis verwendet wird. Während mit der objektorientierten Programmiersprache Java, die Applikation für den Raspberry Pi geschrieben und mit der Datenbanksprache MySQL, die benötigte Datenbank erstellt wird, werden die Textsprachen HTML und CSS für das Erstellen der Webpräsenz und die Programmiersprache PHP für die Kommunikation zwischen der Website und der Datenbank verwendet.

2.3.2.1 Java

1991 gründet Sun Microsystems eine Projektgruppe. Sie soll die zukünftige Ausrichtung der Computer- und Softwareindustrie betrachten und analysieren. Das Ziel dieser Gruppe war die Entwicklung eines programmierbaren Prototyps, welcher für die Steuerung von Haushaltsgeräten zuständig war. Nachdem die Gruppe ein kleines Gerät mit dem Namen „Star Seven“ entwickelt hatte, versuchte man die Vermarktung des Gerätes. Die Vermarktung des Gerätes schlug jedoch fehl. Aus dem Projekt blieb neben dem Maskottchen „Duke“, die Programmiersprache Oak. Da der Name Oak geschützt gewesen war, wurde die Sprache in Java umbenannt. Die Abbildung 5 stellt das Logo von Java dar.

Die offizielle Geburtsstunde von Java ist der 23. Mai 1996, denn an diesem Tag wird Java öffentlich vorgestellt. Ein Jahr später erscheint das erste Java Development Kit (JDK 1.0).



Abbildung 5: Das Java-Logo ²⁵

„Java ist eine vergleichsweise einfach zu erlernende Programmiersprache, die an Popularität immer mehr zunimmt. Dadurch, dass in Java auf die Zeigerarithmetik, die C/C++-Programmierer häufig zur Verzweiflung treibt, verzichtet wurde, entfallen komplexe und fehleranfällige Sprachkonstrukte. Diese Vereinfachung trägt auch enorm zur Stabilität von Programmen bei.“

²⁵ (ohneVerfasser(b), 2015)

In Java ist es vorgesehen, dass man objektorientiert programmiert. Dies fällt allein schon dadurch auf, dass keine Methoden außerhalb von Klassen programmiert werden können. Java bietet eine sehr hohe Sicherheit, da ungewollter Zugriff auf Arbeitsspeicherbereiche und dessen Manipulation durch den Interpreter (Java Virtual Machine, kurz JVM) unterbunden wird.

Der größte Vorteil ist allerdings, dass ein Java-Programm unter jedem Betriebssystem gleich läuft. Der Java-Quellcode wird in einen Bytecode umgewandelt, der nur von der JVM interpretiert wird. Daher muss einzig die JVM auf das jeweilige Betriebssystem angepasst sein. Da Java aber sehr verbreitet ist, stellt es in der Regel kein Problem dar, eine passende JVM für ein Betriebssystem zu finden.²⁶

Zusammengefasst ist Java eine objektorientierte Programmiersprache, die portabel und performant ist. Sie ist einfach in der Verwendung. Aus Java können robuste und dynamische Systeme entwickelt werden, die auch auf verteilten Systemen einwandfrei funktionieren. Außerdem ist Java architekturneutral und kann nebenläufige Prozesse problemlos ausführen.²⁷

2.3.2.2 MySQL

„Der Name MySQL ist auf die Datenbank-Abfragesprache SQL zurückzuführen. Als Anfang der 70er Jahre die ersten Datenbanksysteme auf den Markt kamen, wurde kurze Zeit später eine einheitliche Sprache entwickelt, mit der es möglich sein sollte, Abfragen an jedes beliebige Datenbanksystem zu stellen, ohne jeweils eine neue Sprache lernen und einsetzen zu müssen. Diese Sprache ist SQL, was für „Structured Query Language“ steht und übersetzt „strukturierte Abfragesprache“ bedeutet.“²⁸

„MySQL ist ein SQL-kompatibles Open-Source-Datenbanksystem. Obwohl es ein sehr einfaches System ist, wird es gerade im Internet besonders gern und häufig eingesetzt, weil es nicht wie viele andere Vertreter so sehr überladen und als Open-Source-Software natürlich kostenlos ist.

Wie die meisten Open-Source-Projekte ist auch MySQL unabhängig vom eingesetzten Betriebssystem und somit sowohl für UNIX/Linux als auch für Windows verfügbar.“²⁹

²⁶ (Petri, et al.)

²⁷ Vgl. Ebd., S. 30 – 35.

²⁸ (Lubkowitz, 2007), S. 727.

²⁹ Ebd. S. 48.

2.3.2.3 HTML

HTML (Hypertext Markup Language) wird in das deutsche „Hypertext Auszeichnungssprache“ übersetzt. Es handelt sich um eine Weiterentwicklung von SGML (Structuted Generalized Markup Language), mit welcher sich die Struktur, der Inhalt und das Verhalten eines Dokuments beschreiben bzw. auszeichnen lässt. Es wird kein Programm im eigentlichen Sinne erstellt, weshalb es auch keine Programmier-, sondern eine textbasierte Auszeichnungssprache ist.

Aufgrund wachsender Anforderungen und Veränderungen im World Wide Web, unterlag auch HTML in den vergangenen zehn Jahren einer stetigen Entwicklung und Erneuerung.³⁰

Mit der neuesten Version HTML5 ist es die fünfte Fassung des HTML. Am 28. Oktober 2014 wird HTML5 vom World Wide Web Consortium vorgestellt. Es werden viele Neuerungen vorgestellt. Diese sind unter anderem Audio- und Video-Elemente, mit denen Lieder oder Videos direkt in dem Browser wiedergegeben werden können, ohne einen Plug-In verwenden zu müssen.

Mit den Audio- und Video-Elementen können Lieder und Filme nun direkt im Browser abgespielt werden.³¹



Abbildung 6: HTML5 - Logo³²

³⁰ Vgl. Ebd., S. 109 – 111.

³¹ Vgl. (Jacobs, 2014)

³² Ebd.

2.3.2.4 CSS

„CSS ist die Abkürzung für „Cascading Style Sheets“ und stellt eine Erweiterung zu HTML dar, die von W3C entwickelt wurde. Ihr Ziel ist es, zu den Wurzeln von HTML zurückzukehren und den Inhalt von der Gestaltung zu trennen. Schließlich ist die eigentliche Aufgabe von HTML, Daten und Inhalte zu strukturieren. Diese Aufgabe kann HTML nur schwerlich erfüllen, wenn es mit font-Elementen überladen wird. An dieser Stelle greift nun CSS ein. Es kümmert sich um die optische Gestaltung der Inhalte, während HTML diese nur kennzeichnet und zur Verfügung stellt.“³³

„Über CSS3 kann modernes Webdesign erstellt werden. Es spart dem Entwickler viel Arbeit, da bisher erstellte Grafiken in vielen Fällen nicht mehr notwendig sind. Auch Schatten von Schriften und Bereichen sind im Handumdrehen erstellt. Animationen sind nun über CSS3 möglich und Design kann einfach auf verschiedene Auflösungen und auch für mobile Geräte wie Handys erstellt werden.

CSS3 unterstützt den Entwickler nicht nur bei Design sondern auch für die Entwicklung für verschiedene Geräte und Auflösungen. Web-Apps profitieren von den Möglichkeiten von CSS3 und die Entwicklung für die unterschiedlichen Auflösungen wird möglich. Auf über die neuen Selektoren ergeben sich neue Möglichkeiten.“³⁴

2.3.2.5 JavaScript

„Bei JavaScript handelt es sich um eine Programmiersprache, die von Netscape eingeführt wurde. Daher konnte der Netscape Browser in diesem Bereich auch meistens ein wenig mehr als der von Microsoft. Und auch hier liegt schon die große Gefahr. Es gibt kleine aber feine Unterschiede, ansonsten funktioniert das eingebaute JavaScript-Programm nur bei einigen Surfern.

JavaScript ist also eine Programmiersprache, die im Browser ausgeführt wird und somit Clientseitig ist. Über JavaScript können Inhalte auf einer Webseite geändert, anders dargestellt werden je nach Einstellung bzw. Besucheraktion.“³⁵

³³ (Lubkowitz, 2007), S.247.

³⁴ (Pratzner, 2016)

³⁵ Ebd.

2.3.2.6 PHP

„PHP (Abkürzung für „Personal Homepage Tools“) war ursprünglich lediglich eine Sammlung von mit Perl programmierten Skripts zum Administrieren der Protokolldateien eines Webserver. Rasmus Lerdorf hat diese damals ebenfalls im Internet zur Verfügung gestellt. Kurz darauf wurde PHP jedoch zu einer neuen, eigenständigen Skriptsprache, die die verschiedenen Vorzügen von Perl und andren Sprachen in sich vereinen sollte und diese Aufgabe auch mit Bravour erfüllt.“³⁶

„PHP unterscheidet sich von clientseitigen Sprachen wie JavaScript dadurch, dass der Code auf dem Server ausgeführt wird und dort HTML-Ausgaben generiert, die an den Client gesendet werden. Der Client erhält also nur das Ergebnis der Skriptausführung, ohne dass es möglich ist herauszufinden, wie der eigentliche Code aussieht.“³⁷

³⁶ (Lubkowitz, 2007), S. 48.

³⁷ (Achour, et al.)

2.3.3 Hardware

Nachdem im Kapitel 2.3.2 die Programmiersprachen erläutert wurden, erfolgt in dem Kapitel 2.3.3 die Aufzählung der verwendeten Hardware. Hierbei werden Vergleiche gemacht, Besonderheiten erwähnt und die entstandenen Kosten erläutert.

2.3.3.1 Raspberry Pi

„Der Raspberry Pi ist ein [Einplatinencomputer], welcher aus lediglich einer Platine besteht und erstmals 2012 auf den Markt kam. Das Gerät ohne Gehäuse wurde ursprünglich dafür konzipiert, Leute an Hardwarekenntnisse und Programmierfähigkeiten heranzuführen, weswegen von Beginn an ein niedriger Preis angestrebt wurde. Die Idee fand schnell Interessenten, was zu einer heute sehr ausgeprägten Community rund um DIY-Projekte oder Basteleien mit der winzigen Hardware führte. Inzwischen gibt es mehrere Versionen und unterschiedliches Zubehör des mehrfach ausgezeichneten Raspberry Pi. Seit Modell 3 sind WLAN und Bluetooth bereits integriert.“³⁸

In der Planungsphase wurde für die Durchführung der Thesis-Aufgabenstellung, das Raspberry Pi Modell 2 B geplant. Dazu wurde ein Raspberry PI Modell 2B Quad Core Starter Bundle gekauft. Der Preis des Bundles beläuft sich auf 74,95 Euro und beinhaltete folgende Komponente:

- Raspberry PI Modell 2 B, mit einem 900 MHz Quad Core Prozessor und 512 MB RAM-Speicher
- 8 GB SD-Karte mit vorinstalliertem NOOBS-System
- 1x Gehäuse
- Netzteil mit einem USB-Kabel
- 1x 2m langes HDMI-Kabel
- 1x LAN-Kabel
- 1x WLAN USB-Stick³⁹

Nach dem Zusammenbau wurde Raspbian auf dem System installiert. Man musste jedoch feststellen, dass die WLAN-Verbindung nicht stabil gewesen ist. Für das Projekt aber wird eine dauerhafte Internetverbindung vorausgesetzt. Die mögliche Lösung war, die Verbindung über ein LAN-Kabel zu errichten. Diese Idee wurde jedoch verworfen, da ein weiteres Kabel an dem Raspberry Pi nicht erwünscht wurde.

³⁸ (Golem.de)

³⁹ Vgl. (Pi)

Aus diesem Grund wurde für die Thesis ein Raspberry Pi 3 Modell B gewählt. Dieses Modell hat ein integriertes WLAN-Modul, wodurch die Internetverbindung dauerhaft möglich ist. Der Raspberry Pi 3 B wurde über Amazon mit einem Preis von 45 Euro erworben und hat folgende Eigenschaften:

- Ein Quad Core Prozessor mit 1,2 GHz
- 1 GB DDR2 RAM
- Integriertes WLAN-Modul
- Integriertes Bluetooth-Modul⁴⁰

Damit der Raspberry Pi geschützt und gekühlt wird, wurden drei Kühlelemente und ein Gehäuse gekauft. Die Elemente wurden, wie auch der Raspberry Pi über Amazon erworben. Das Gehäuse hatte ein Preis von 6,29 Euro⁴¹, die Kühlelemente ein Preis von 3,99 Euro⁴². Nach dem Zusammenbau sieht das Ergebnis, wie in der Abbildung 7 und 8 dargestellt, aus.



Abbildung 7: Raspberry Pi 3 Modell B Nr.1



Abbildung 8: Raspberry Pi 3 Modell B Nr.2

⁴⁰ Vgl. (Raspberry)

⁴¹ Vgl. (SB Components)

⁴² Vgl. (Aukru)

2.3.3.2 USB Netzteil

Das USB-Netzteil wird aus dem Raspberry PI Modell 2B Quad Core Starter Bundle verwendet. Da alle Raspberry Pi über Micro-USB mit Strom versorgt werden, musste hierfür kein neues Netzteil erworben werden. Der Preis von 8,99 Euro⁴³ wird anhand eines standartmäßigen Micro-USB Netzteils festgelegt. Das Netzteil und USB-Kabel sehen wie in Abbildung 9 und 10 dargestellt aus.



Abbildung 9: USB-Netzteil Nr.1



Abbildung 10: USB-Netzteil Nr.2

⁴³ Vgl. (Valadur® Supplies)

2.3.3.3 HDMI-Kabel

Das HDMI-Kabel wurde ebenso über das Bundle erworben, weshalb die Berechnung des Preises schwer fällt. Deshalb wird der Endpreis mit 3,97 Euro⁴⁴ anhand eines einfachen HDMI-Kabels ermittelt.



Abbildung 11: HDMI-Kabel

2.3.3.4 SD-Karte

Die SD-Karte wurde in Media Markt erworben. Mit dem Preis von 12,99 Euro⁴⁵ ist die SD-Karte von Samsung mit 32 GB relativ günstig.



Abbildung 12: SD-Karte mit 32 GB

⁴⁴ Vgl. (kabelbude.eu)

⁴⁵ Vgl. (Media Markt)

2.3.3.5 Barcode-Scanner

Der Barcode-Scanner wurde ebenso über Amazon für einen Preis von 22,99⁴⁶ Euro erworben. Der Scanner funktioniert wie eine Tastatur. Nach dem Einscannen des Barcodes wird dieser gleich abgeschickt. Diese Funktion erweist sich als sehr vorteilhaft, da dadurch die Programmierung der Anwendung erleichtert wird. In der Abbildung 12 wird der Barcode-Scanner abgebildet.



Abbildung 13: Barcode-Scanner

2.3.3.6 Raspberry Pi Touch Screen mit 2x13 Pin Header-Verlängerung

Die JaRasPi-Anwendung soll vom Benutzer gestartet und angehalten werden können. Damit an dem Raspberry Pi kein Bildschirm angeschlossen werden muss, wurde ein Touch Screen (22 Euro)⁴⁷ erworben. Bei der Touch Screen-Lieferung wurde zusätzlich eine Installations-DVD beigelegt. Diese beinhaltet das OS-Raspbian Version Jessie und den vorinstallierten Treiber für das Display.

Da der Raspberry Pi aus Schutzgründen ein Gehäuse verfügt, muss der Touch Screen über eine Verlängerung (4,11 Euro)⁴⁸ an den Pi angeschlossen werden.

⁴⁶ (TaoTronics)

⁴⁷ (Kuman)

⁴⁸ (Adafruit)



Abbildung 14: Raspberry Pi Touch Screen mit 2x13 Pin Header-Verlängerung

2.3.3.7 Kamera Modul für Raspberry Pi mit Kameragehäuse

Eines der Aufgaben der JaRasPi-Anwendung ist das Erstellen von Abbildungen, beispielsweise einer Fruchtschale oder eines Lebensmittelschranks. Diese Funktion wird mit Hilfe einer Raspberry Pi Kamera (29,99 Euro)⁴⁹ ermöglicht. Da die Kamera keine Hülle besitzt und es eine Gefahr für ESD(Elektrostatische Entladung)-Schaden besteht, musste für die Kamera ein Gehäuse (7,77 Euro)⁵⁰ miterworben werden.



Abbildung 15: Kamera Modul für Raspberry Pi mit Kameragehäuse Nr.1

⁴⁹ (Raspberry)

⁵⁰ (sb.component)

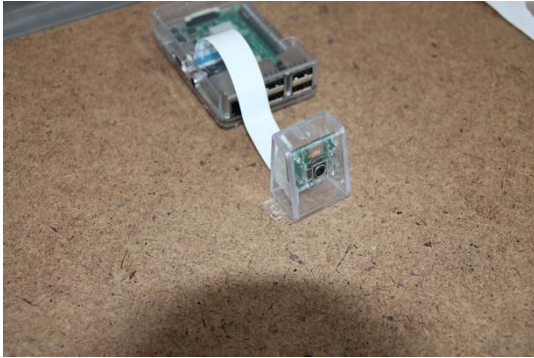


Abbildung 16: Kamera Modul für Raspberry Pi mit Kameragehäuse Nr.2

2.3.3.8 Zusammenbau des Endsystems

Das zusammengebaute System kann in der Abbildung 16 betrachtet werden. Mit einem Gesamtpreis von 168,09 Euro fällt das System recht kostengünstig aus.

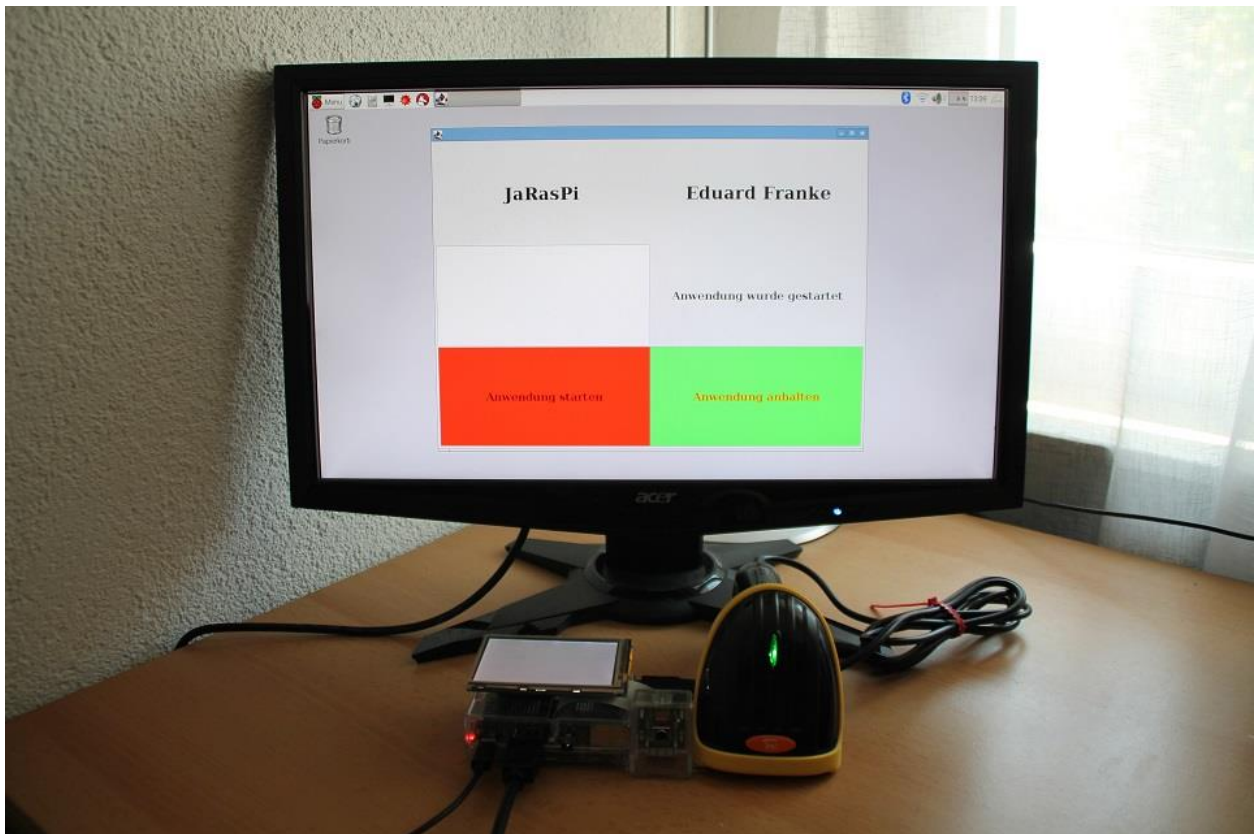


Abbildung 17: Gesamtaufbau des Systems

2.4 Phasenplanung und Meilensteine des Projektes

Zeitraum	Meilensteine
01.03.2016 13.04.2016	In dem Zeitraum vom 01. März bis 01. April wird die Projektplanung durchgeführt. Hierbei werden Ideen für das Projekt gesammelt. Nachdem das Thema endgültig steht wird die Literaturrecherche angegangen. Außerdem werden die notwendigen Komponenten gekauft, zusammengesetzt und auf die Funktion hin getestet.
14.04.2016 05.06.2016	In dem Zeitraum vom 14. April bis einschließlich 05. Juni wird das Kapitel 2 und Kapitel 3 bearbeitet und abgeschlossen. Im Kapitel 3 wird der Aufbau und die Logik des Systems dargestellt. Deshalb muss dieses Kapitel zuerst abgearbeitet werden, erst dann kann die Programmierung begonnen werden.
06.06.2016 17.07.2016	In dem Zeitraum vom 06. Juni bis einschließlich 17. Juli wird die Programmierung der Applikation durchgeführt.
18.07.2016 24.07.2016	Nachdem die Programmierung der Anwendung abgeschlossen ist, wird in dem Zeitraum vom 18. Juli bis zum 24. Juli das System getestet. Sollte der Test schnell und erfolgreich sein, wird die Fertigstellung der Thesis in Angriff genommen.
25.07.2016 07.08.2016	In den zwei Wochen vom 25. Juli bis einschließlich 07. August wird die Thesis fertiggestellt.
08.08.2016 15.08.2016	Letzter Akt: Vom 08. August bis zum 15. August wird das Drucken und Verschicken der fertiggestellten Thesis durchgeführt.

Tabelle 1: Phasenplanung und Meilensteine

3 Theoretischer Bezugsrahmen

In den nächsten Kapiteln wird der theoretische Bezugsrahmen erläutert. Dabei werden im Kapitel 3.1, die Inhalte eines Pflichtenheftes und im Kapitel 3.2, die Inhalte der Systemarchitektur abgearbeitet. Die Kenntnisse von Software Engineering aus dem vierten Semester und das Buch von Bernd Oestereich, Analyse und Design mit UML 2.3, werden dabei sehr hilfreich sein. Die Modellierung der Diagramme und Strukturen wird mit dem Softwaremodellierungswerkzeug Enterprise Architekt von Sparx Systems durchgeführt.

Da in der Entwicklungsphase viele neue Ideen gesammelt, manche jedoch weggelassen werden, müssen einige Punkte im Kapitel 3 überarbeitet werden.

3.1 Anforderungsdefinition

3.1.1 Systemkontext

Nachdem im Kapitel 2 die Systemidee und die Zielsetzung erläutert wurden, ist der nächste Schritt die Aufnahme des Systemkontextes. Folgende Punkte sind hierbei wichtig:

- Das System wird als eine Klasse in einem Diagramm dargestellt
- Alle Fremdsysteme mit denen das System arbeiten wird, müssen ermittelt werden
- Die Benutzer des Systems werden als Akteure dargestellt
- Die Beziehungen zwischen den Akteuren/ Fremdsystemen und dem zukünftigen System müssen modelliert werden⁵¹

Das Ergebnis sieht wie in der Abbildung 17 dargestellt aus.

⁵¹ Vgl. (Oestereich, 2009), S. 104 – 105.

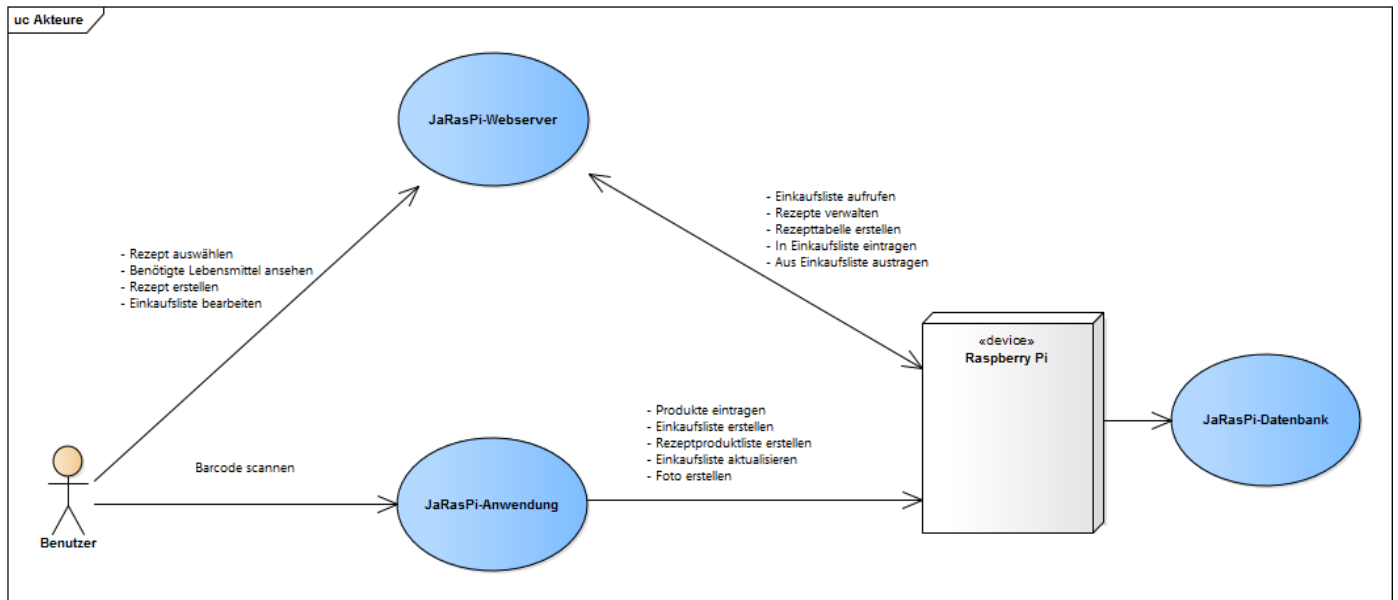


Abbildung 18: Systemkontext der JaRasPi-Anwendung

Das zu konstruierende System hat einen Akteur und zwar den Benutzer, der das System verwenden wird. Weiterhin gibt es den Raspberry Pi auf dem das Hauptprogramm, der Webserver und eine MySQL-Datenbank laufen.

3.1.2 Kurzbeschreibung der Akteure

Akteur	Kurzbeschreibung
Benutzer	<ul style="list-style-type: none"> • Scannt die benötigten Artikel mit dem Barcode-Scanner ein • Ruft die Einkaufsliste auf • Entfernt ein Produkt aus der Einkaufsliste • Fügt ein Produkt der Einkaufsliste zu • Wählt Rezepte aus • Erstellt Rezepte • Kann den Inhalt einer Fruchtschale anzeigen lassen

Tabelle 2:Kurzbeschreibung der Akteure

3.1.3 Kurzbeschreibung der Use Cases

Das Kapitel 3.1.3 beschreibt alle vorhandenen Use Cases in Kurzform. Sie wurden gegenüber der ersten Version überarbeitet, da in der Entwicklungsphase neue Ideen hinzukamen und andere hingegen geändert wurden.

ID	Use Case	Kurzbeschreibung
UC-1	Barcode scannen	<p>Der Benutzer scannt den Barcode des verbrauchten/neuen Produktes ein. Dadurch wird die JaRasPi-Anwendung angestoßen, den Barcode zu identifizieren, zu übersetzen und das verbrauchte/neue Produkt in die Produktliste als verbraucht/gekauft zu speichern.</p> <p>Bei vorhandenen Produkten wird der Bestand um 1 verringert.</p>
UC-2	Weiteres Produkt hinzufügen	Drückt der Benutzer auf der JaRasPi Oberfläche den Button „Weiteres Produkt hinzufügen“ und scannt ein Produkt ein, wird dieses nicht ausgescannt, sondern der Bestand erhöht sich um 1 Stück.
UC-3	Rezeptauswahl kontrollieren	Der Benutzer wählt ein Rezept auf der Webseite aus. Das Programm kontrolliert, welches Rezept ausgewählt wurde und füllt anhand der Daten die Rezeptproduktliste.
UC-4	Einkaufsliste erstellen	<p>Das Programm ruft die Klasse, in der die Einkaufsliste erstellt wird auf. Als nächstes wird in jeder Tabelle überprüft, ob ein Produkt als nicht vorhanden markiert ist und ob es Rezepte gibt, die ausgewählt worden sind.</p> <p>Wurde ein Produkt gefunden, wird es in die Einkaufsliste eingetragen.</p>

UC-5	Einkaufsliste aktualisieren	<p>Das Programm ruft die Klasse, in der die Einkaufsliste aktualisiert wird. Dabei kontrolliert das Programm, ob ein Produkt in die Datenbank als „vorhanden“ eingetragen wurde.</p> <p>Ist das der Fall, wird das Produkt aus der Einkaufsliste entfernt.</p>
UC-6	Foto erstellen	<p>Die JaRasPi-Anwendung erstellt in bestimmten Abständen ein Foto vom Inhalt eines Lebensmittelschranks oder einer Fruchtschale und speichert es auf dem Raspberry Pi.</p>
UC-7	Einkaufsliste auswählen	<p>Der Benutzer wählt auf der Website „Einkaufsliste anzeigen“ aus. Der Webserver greift mit einem lesenden Zugriff auf die Datenbank zu.</p> <p>Nachdem die Daten ausgelesen wurden, werden diese dem Benutzer in Form einer digitalen Einkaufsliste bereitgestellt.</p>
UC-8	Inhalt anzeigen	<p>Der Benutzer wählt die Option „Inhalt anzeigen“ aus.</p> <p>Der Webserver greift auf den Ordner mit den erstellten Fotos in dem Raspberry Pi zu und stellt das Foto dem Benutzer bereit.</p>
UC-9	Rezept auswählen	<p>Der Benutzer wählt auf der Website ein Rezept aus.</p> <p>Der Webserver liest die benötigten Daten aus und markiert in der Rezept-Tabelle das ausgewählte Rezept.</p>

UC-10	Rezept erstellen	<p>Der Benutzer wählt auf der Website „Rezept erstellen“ aus.</p> <p>Es erscheinen mehrere Eingabefelder, in die der Benutzer die benötigten Daten einträgt und das Rezept erstellt.</p>
UC-11	Produkt entfernen	<p>Wurde „Produkt entfernen“ ausgewählt, kann der Benutzer manuell Produkte aus der Einkaufsliste entfernen. Wurde ein Produkt entfernt, wird der Benutzer darüber informiert, dass das Austragen erfolgreich gewesen ist.</p>
UC-12	Produkt hinzufügen	<p>Wurde „Produkt hinzufügen“ ausgewählt, kann der Benutzer manuell Produkte in die Einkaufsliste hinzufügen. Wurde ein Produkt hinzugefügt, wird der Benutzer darüber informiert, dass das Eintragen erfolgreich gewesen ist.</p>

Tabelle 3: Kurzbeschreibung der Use Cases

3.1.4 Anwendungsfalldiagramm

„Ein Anwendungsfalldiagramm beschreibt die Zusammenhänge zwischen einer Menge von Anwendungsfällen und den daran beteiligten Akteuren.“⁵² Das in der Abbildung 18 dargestellte Anwendungsfalldiagramm stellt die Systemgrenzen des zu erstellenden Systems dar.

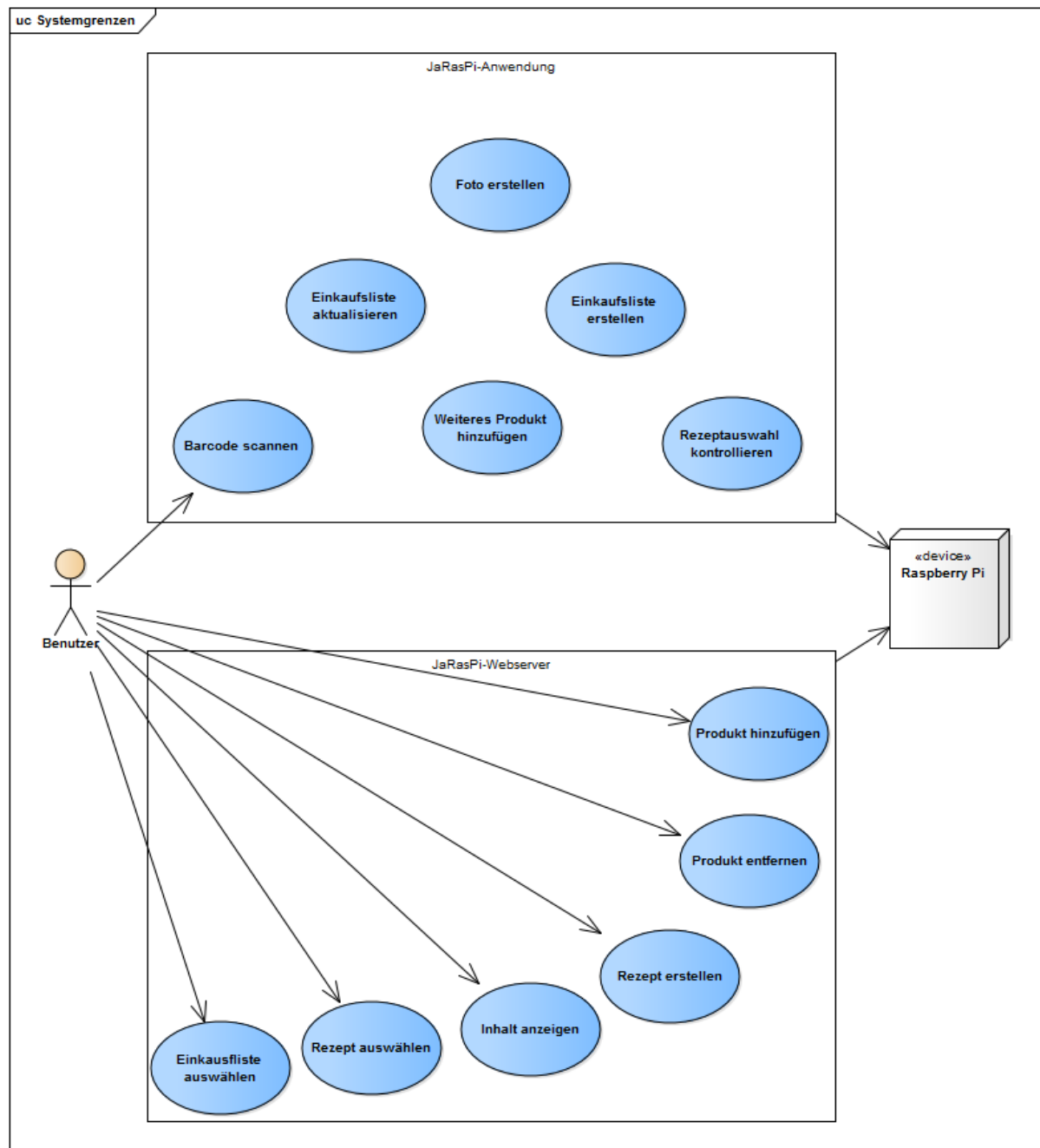


Abbildung 19: Systemgrenzen

⁵² (Oestereich, 2009), S. 245.

3.1.5 Fachklassenmodell

3.1.5.1 Klassendiagramm

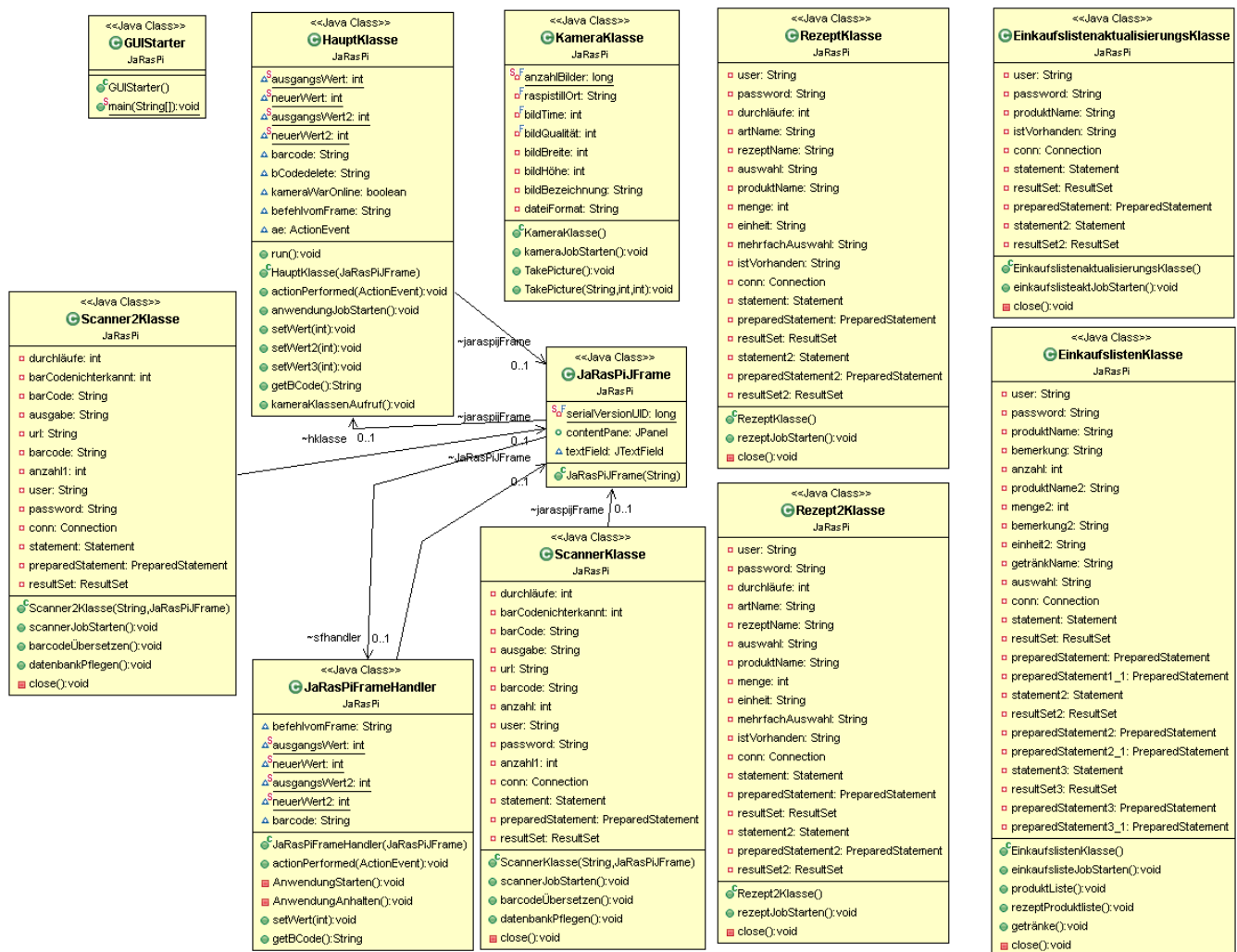


Abbildung 20: Klassendiagramm der JaRasPi-Anwendung

3.1.5.2 Klassenmodell

„Ein Klassenmodell beschreibt, welche Klassen existieren und in welchen Beziehungen sie zueinander stehen.“⁵³

⁵³ (Oestereich, 2009), S. 273.

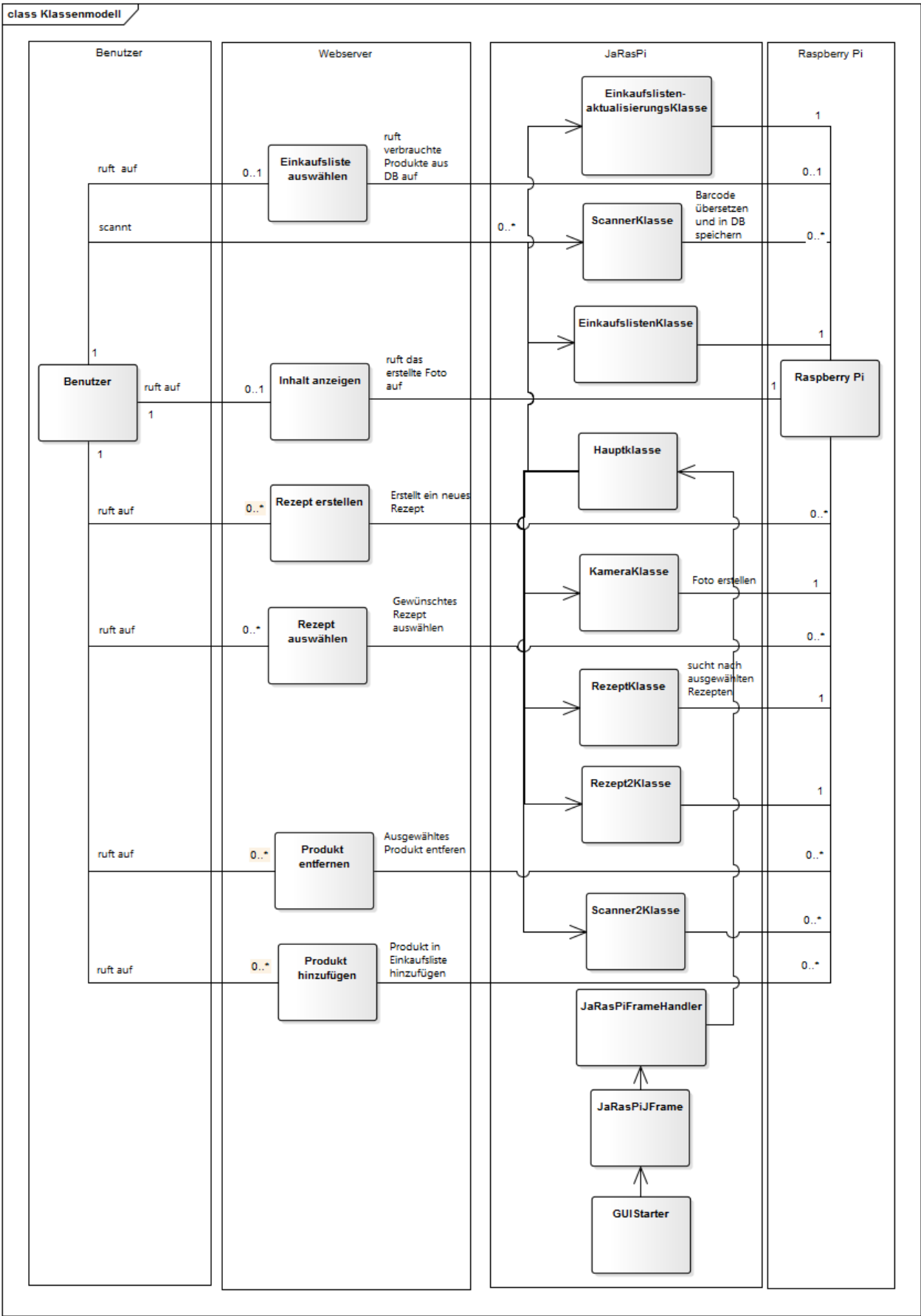


Abbildung 21: Klassenmodell

3.1.5.3 Klassenbeschreibung/Attributbeschreibung

Da die meisten Klassen eine enorme Anzahl der Attribute aufweisen, werden die Attribute nicht explizit aufgeführt. Die Klassen mit ihren Attributen können aus dem Klassendiagramm im Kapitel 3.1.5.1 entnommen werden.

Klasse ID	Klasse Bezeichnung	Klasse Beschreibung
ELAK1	EinkaufslistenaktualisierungsKlasse	<p>Das Programm ruft die Einkaufsliste auf und kontrolliert in der Produktliste, ob das Produkt XY vorhanden ist oder nicht.</p> <p>Sind Produkte vorhanden, werden diese aus der Einkaufsliste entfernt.</p>
ELK1	EinkaufslistenKlasse	<p>Das Programm ruft in der EinkaufslistenKlasse die Tabellen nacheinander auf.</p> <p>Zuerst wird die Produktliste aufgerufen und kontrolliert, ob Produkte mit „nichtVorhanden“ markiert sind und fügt sie in die Einkaufsliste ein.</p> <p>Danach wird die Rezeptproduktliste aufgerufen. Hier werden alle Artikel in die Einkaufsliste hinzugefügt.</p> <ul style="list-style-type: none"> • Wichtig: Sind gleiche Produkte in der Einkaufsliste vorhanden, wird nur die Menge zusammenaddiert. <p>Als letztes wird die Getränke-Tabelle aufgerufen. Sind Produkte ausgewählt worden, werden diese in die Einkaufsliste hinzugefügt.</p>
GS1	GUIStarter	Die Klasse GUIStarter, startet die JaRasPiJFrame Klasse.
HK1	HauptKlasse	In der HauptKlasse werden die einzelnen Unterklassen (siehe Klassenmodell) nacheinander aufgerufen.
JFH1	JaRasPiFrame-Handler	Die Klasse JaRasPiFrameHandler ist für die Interaktion des Benutzers und der GUI zuständig.
JJF1	JaRasPiJFrame	Die Klasse JaRasPiJFrame erzeugt die Oberfläche, die der

		Benutzer beim Starten der Anwendung sieht.
KK1	KameraKlasse	In bestimmten Zeitabständen wird die KameraKlasse aufgerufen, um mit dem Raspberry Pi Kameramodul ein Foto zu erzeugen.
R2K1	Rezept2Klasse	<p>Die Rezept2Klasse überprüft, ob in der Rezept2-Tabelle Rezepte vorhanden sind, die mit „ausgewaehlt“ markiert wurden.</p> <p>Ist das der Fall, wird das jeweilige Rezept aufgerufen und alle benötigten Artikel in die Rezeptproduktliste hinzugefügt.</p> <ul style="list-style-type: none"> • Wichtig: Bevor die Produkte in die Rezeptproduktliste hinzugefügt werden, wird kontrolliert, ob das Produkt in der Produktliste vorhanden ist oder nicht.
RK1	RezeptKlasse	<p>Die RezeptKlasse überprüft, ob in der Rezept-Tabelle Rezepte vorhanden sind, die mit „ausgewaehlt“ markiert wurden.</p> <p>Ist das der Fall, wird das jeweilige Rezept aufgerufen und alle benötigten Artikel in die Rezeptproduktliste hinzugefügt.</p> <ul style="list-style-type: none"> • Wichtig: Bevor die Produkte in die Rezeptproduktliste hinzugefügt werden, wird kontrolliert, ob das Produkt in der Produktliste vorhanden ist oder nicht. <p>Die RezeptKlasse unterscheidet sich von der Rezept2Klasse in der Hinsicht, dass die RezeptKlasse, für die vom Programmierer hinzugefügten Rezepte zuständig ist. Während die Rezept2Klasse, die vom Benutzer erstellten Rezepte verwaltet.</p>
SK1	ScannerKlasse	<p>Die ScannerKlasse ist das Herzstück der JaRasPi-Anwendung. Nachdem der Barcode eingescannt wurde, wird der Code hier übersetzt und in die Produktliste hinzugefügt.</p> <ul style="list-style-type: none"> • Wichtig: In der ScannerKlasse wird jedes neue Produkt als „nichtVorhanden“ und mit einer Anzahl von 1 in die Produktliste aufgenommen • Bei bereits aufgenommenen Produkten wird die Anzahl um 1 verringert. • Ist der Mindestbestand von 1 erreicht, wird das Pro-

		dukt als „nichtVorhanden“ markiert und in die Einkaufsliste aufgenommen.
S2K1	Scanner2Klasse	<p>Die Scanner2Klasse ist das Herzstück der JaRasPi-Anwendung. Nachdem der Barcode eingescannt wurde, wird der Code hier übersetzt und in die Produktliste hinzugefügt.</p> <ul style="list-style-type: none">• Wichtig: In der Scanner2Klasse wird jedes neue Produkt als „nichtVorhanden“ und mit einer Anzahl von 1 in die Produktliste aufgenommen• Bei bereits aufgenommenen Produkten wird die Anzahl um 1 erhöht.

Tabelle 4: Klassenbeschreibung/Attributbeschreibung

3.1.6 Systemfunktionalität

Nachdem im Kapitel 3.1.3 die Use Cases kurz erläutert wurden, werden diese in den nächsten Kapiteln ausführlich abgearbeitet und die dazugehörigen Systemablaufmodelle modelliert.

Bevor die Use Cases beschrieben werden, müssen folgende Punkte beachtet werden:

- Für jeden Geschäftsanwendungsfall sollen die Auslöser, Vorbedingungen und die eingehenden Informationen identifiziert werden
- Es soll zwischen stabilen und sich ändernden Anforderungen unterschieden werden
- Für jeden Use Case sollen Ergebnisse, Nachbedingungen und wenn notwendig, die ausgehenden Informationen identifiziert werden
- Es ist wichtig mit einem Anwendungsfall nur genau einen fachlichen Sachverhalt zu beschreiben
- Ein Anwendungsfall soll ausschließlich die fachliche Essens beschreiben
- Die Geschäftsanwendungsfälle sollen keine technologische Lösungen oder Rahmenbedingungen enthalten⁵⁴

Bei der Modellierung von Systemablaufmodellen ist wichtig, dass jeder Anwendungsfallschritt in einen oder mehrere elementare Schritte zerlegt wird. Weiterhin ist es wichtig, dass für jeden Schritt alle vorgesehenen Ausnahmen, Abbrüche und Verzweigungsmöglichkeiten modelliert werden.⁵⁵

Damit der Überblick erhalten bleibt, wird nach jedem Use Case das dazugehörige Systemablaufmodell folgen.

⁵⁴ Vgl. (Oestereich, 2009), S. 112 – 119.

⁵⁵ Vgl. Ebd., S. 142 – 152.

3.1.6.1 Use Case Barcode scannen

Name	Barcode scannen
Kurzbeschreibung	<p>Der Benutzer scannt den Barcode des verbrauchten/neuen Produktes ein. Dadurch wird die JaRasPi-Anwendung angestoßen, den Barcode zu identifizieren, zu übersetzen und das verbrauchte/neue Produkt in die Produktliste als verbraucht/gekauft zu speichern.</p> <p>Bei vorhandenen Produkten wird der Bestand um 1 verringert.</p>
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer verbraucht/kauft ein Produkt und scannt den Barcode des Artikels mithilfe des Barcode-Scanners ein.
Vorbedingungen	-
Eingehende Informationen	Barcode
Ergebnisse	<p>Der Barcode wurde übersetzt und in die Produktliste als:</p> <ul style="list-style-type: none"> • ein neues Produkt eingetragen oder • das vorhandene Produkt wurde um Menge 1 verringert
Nachbedingungen	In der Produktliste ist der Eintrag eines neuen Produktes vorhanden oder die Menge des vorhandenen Produktes wurde nach dem Scannen um 1 verringert.
Ablauf	<ol style="list-style-type: none"> 1. Scann Vorgang starten Der Benutzer nimmt den Barcodescanner und scannt das verbrauchte/neue Produkt ein. 2. Barcode übersetzen Der eingescannte Barcode wird in der JaRasPi-Anwendung aufgenommen und mit Hilfe einer Webseite übersetzt. 3. Übersetzten Barcode in die Produktliste eintragen Nachdem der Barcode übersetzt wurde, wird das Produkt in die Datenbank eingetragen. Dabei werden folgende Parameter eingetragen oder geändert: <ul style="list-style-type: none"> • Barcode

	<ul style="list-style-type: none"> • Artikelname • Ist vorhanden • Anzahl • Bemerkung
Risiko	Hohes Risiko
Verbindlichkeit, Priorität	Hohe Priorität, unverzichtbar

Tabelle 5: Use Case 1 Barcode scannen

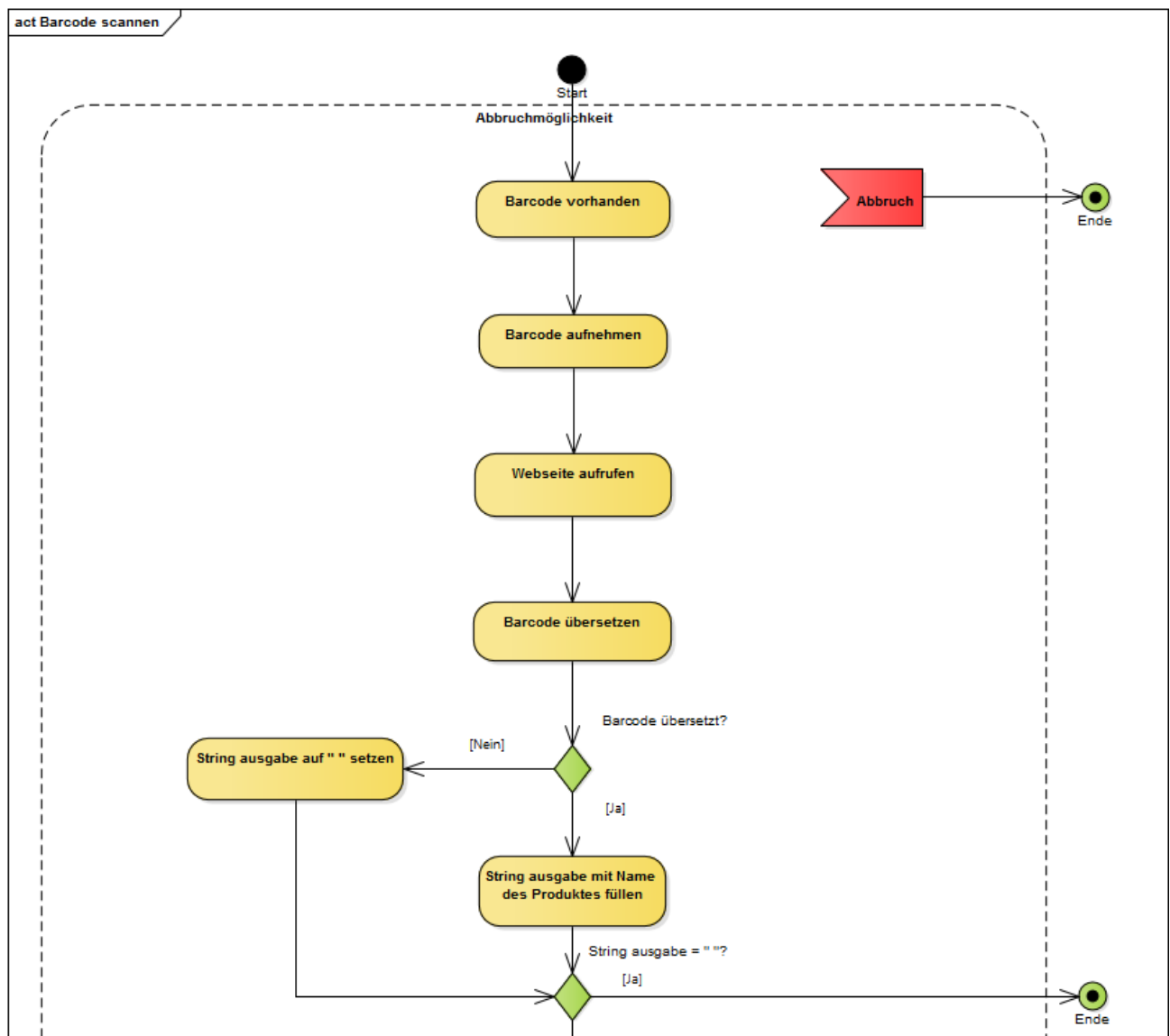


Abbildung 22: Systemablaufmodell Barcode scannen Nr.1

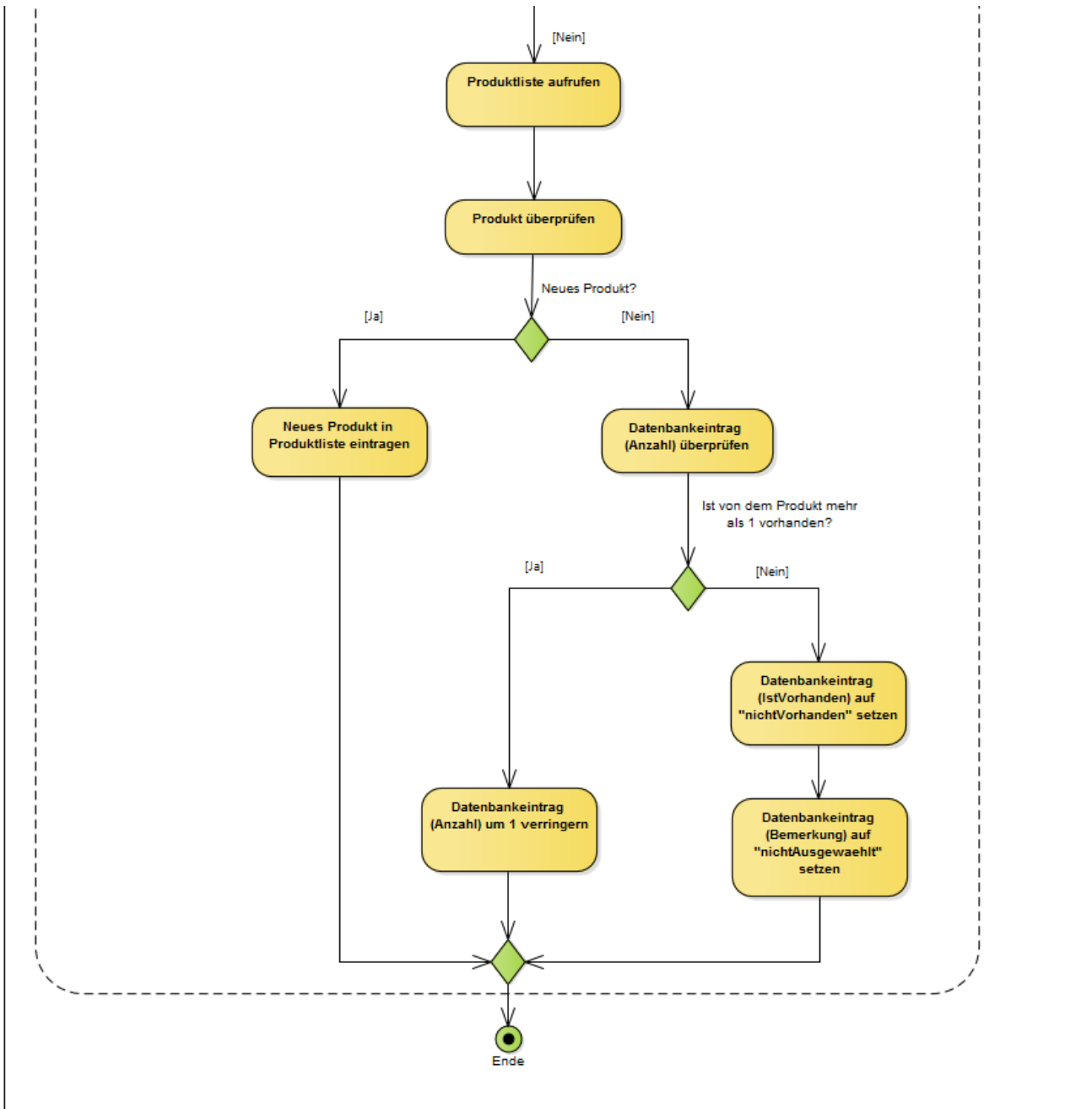


Abbildung 23: Systemablaufmodell Barcode scannen Nr.2

3.1.6.2 Use Case Weiteres Produkt hinzufügen

Name	Weiteres Produkt hinzufügen
Kurzbeschreibung	Drückt der Benutzer auf der JaRasPi Oberfläche den Button „Weiteres Produkt hinzufügen“ und scannt ein Produkt ein, wird dieses nicht aus- gescannt, sondern der Bestand erhöht sich um 1 Stück.
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer kauft ein Produkt und scannt den Barcode des Artikels mit Hilfe des Barcode-Scanners ein.
Vorbedingungen	-
Eingehende Informationen	Barcode
Ergebnisse	Der Barcode wurde übersetzt und in die Produktliste als: <ul style="list-style-type: none"> • ein neues Produkt eingetragen oder • das vorhandene Produkt wurde um Menge 1 erhöht
Nachbedingungen	In der Produktliste ist der Eintrag eines neuen Produktes vorhanden oder die Menge des vorhandenen Produktes wurde nach dem Scannen um 1 verringert.
Ablauf	<ol style="list-style-type: none"> 1. Scann Vorgang starten Der Benutzer nimmt den Barcodescanner und scannt das neu gekaufte Produkt ein. 2. Barcode übersetzen Der eingescannte Barcode wird in der JaRasPi-Anwendung aufgenommen und mit Hilfe einer Webseite übersetzt. 3. Übersetzten Barcode in die Produktliste eintragen Nachdem der Barcode übersetzt wurde, wird das Produkt in die Datenbank eingetragen. Dabei werden folgende Parameter eingetragen oder geändert: <ul style="list-style-type: none"> • Barcode • Artikelname • Ist vorhanden • Anzahl

	<ul style="list-style-type: none"> Bemerkung
Risiko	Hohes Risiko
Verbindlichkeit, Priorität	Hohe Priorität, unverzichtbar

Tabelle 6: Use Case 2 Weiteres Produkt hinzufügen

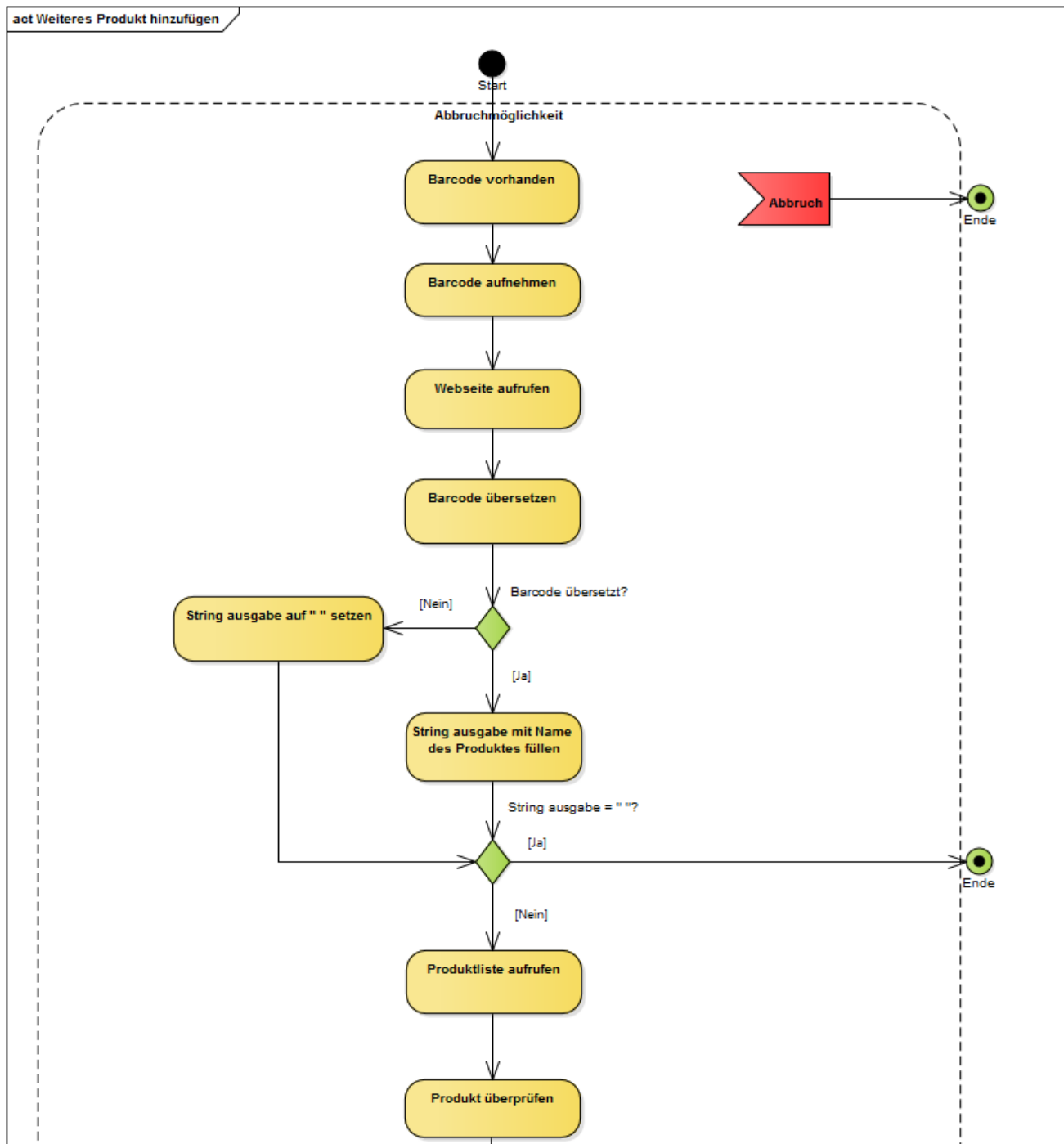


Abbildung 24: Systemablaufmodell Weiteres Produkt hinzufügen Nr.1

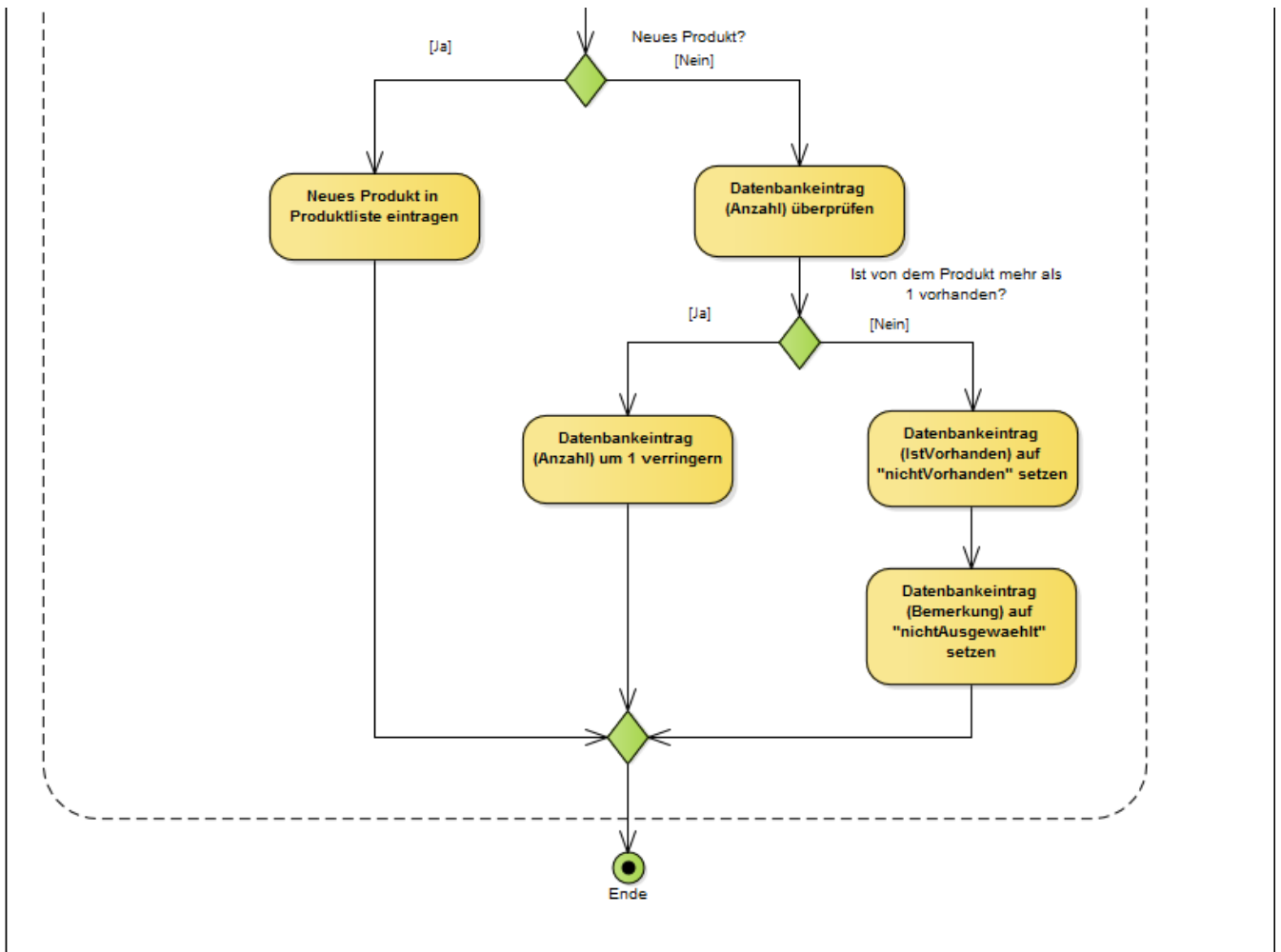


Abbildung 25: Systemablaufmodell Weiteres Produkt hinzufügen Nr.2

3.1.6.3 Use Case Rezeptauswahl kontrollieren

Name	Rezeptauswahl kontrollieren
Kurzbeschreibung	Der Benutzer wählt ein Rezept auf der Webseite aus. Das Programm kontrolliert, welches Rezept ausgewählt wurde und füllt anhand der Daten die Rezeptproduktliste.
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer wählt ein Rezept auf der Webseite aus.
Vorbedingungen	-
Eingehende Informationen	Eintrag in die Rezept-Tabelle
Ergebnisse	Die Produkte aus dem ausgewählten Rezept werden, falls in der Produktliste nicht vorhanden, in die Rezeptproduktliste eingetragen.
Nachbedingungen	<ul style="list-style-type: none"> • Die Rezeptproduktliste ist mit benötigten Produkten gefüllt • Das Rezept wurde mit „nichtAusgewaehlt“ markiert
Ablauf	<ol style="list-style-type: none"> 1. Rezept-Tabelle überprüfen Das Programm überprüft, ob ein Rezept ausgewählt wurde. 2. Rezeptproduktliste aufrufen Das ausgewählte Rezept wird aufgerufen. 3. Rezeptproduktliste pflegen Nachdem das Rezept aufgerufen wurde, wird jedes einzelne Produkt in dem Rezept durchlaufen und falls dieses in der Produktliste nicht vorhanden ist, wird es in die Rezeptproduktliste eingetragen. 4. Aus Rezept-Tabelle austragen Nachdem das Rezept abgearbeitet wurde, wird das ausgewählte Rezept aus der Rezept-Tabelle abgewählt, in dem es mit „nichtAusgewaehlt“ markiert wird.
Risiko	Hohes Risiko
Verbindlichkeit, Priorität	Hohe Priorität, unverzichtbar

Tabelle 7: Use Case 3 Rezeptauswahl kontrollieren

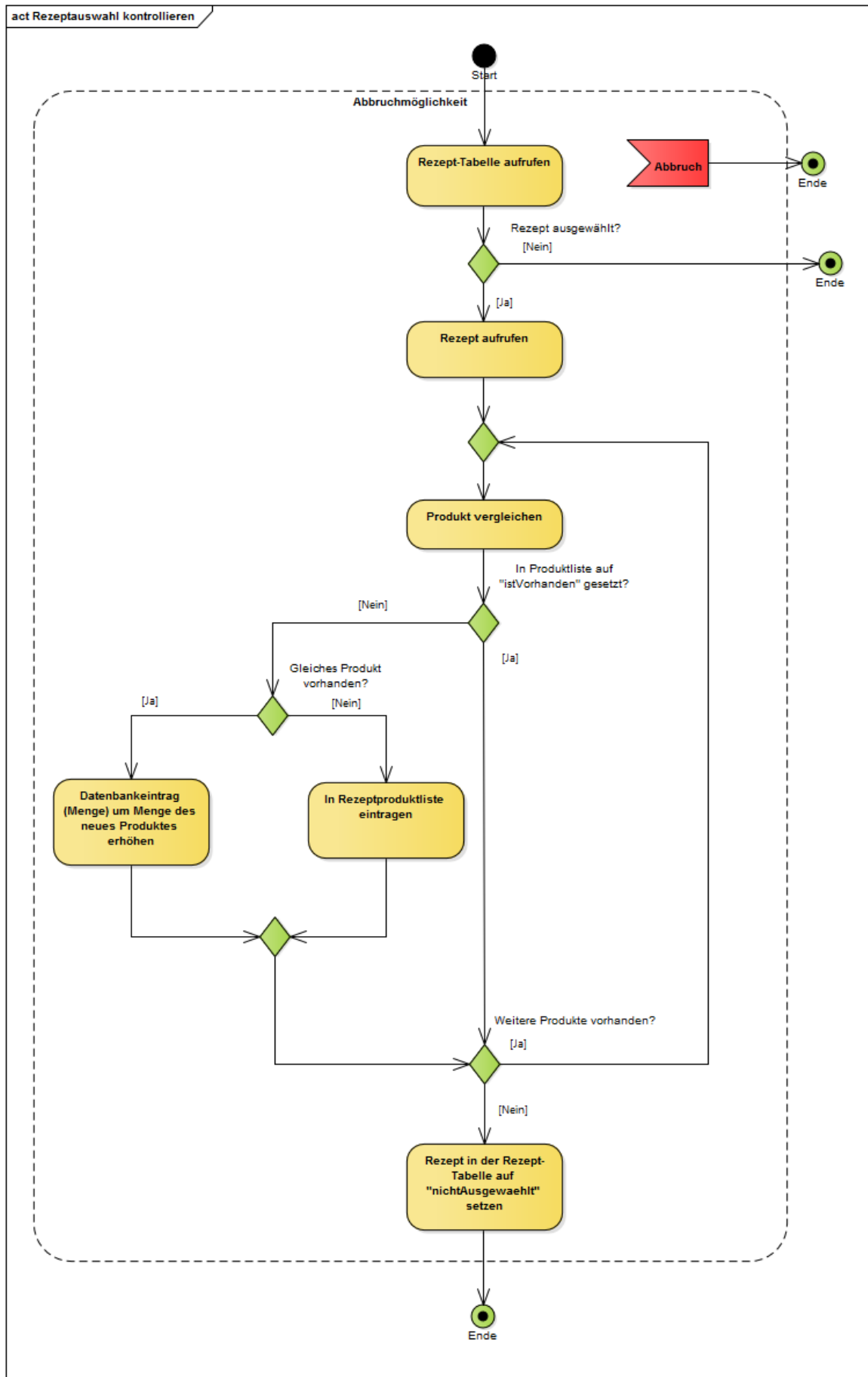


Abbildung 26: Systemablaufmodell Rezeptauswahl kontrollieren

3.1.6.4 Use Case Einkaufsliste erstellen

Name	Einkaufsliste erstellen
Kurzbeschreibung	<p>Das Programm ruft die Klasse, in der die Einkaufsliste erstellt wird auf. Als nächstes wird in jeder Tabelle nachgesehen, ob ein Produkt als nicht vorhanden markiert ist und ob es Rezepte gibt, die ausgewählt worden sind.</p> <p>Wurde ein Produkt gefunden, wird es in die Einkaufsliste eingetragen.</p>
Akteure	-
Auslöser	<ul style="list-style-type: none"> • Rezept wurde ausgewählt • Getränk wurde ausgewählt • Produkt ist in der Produktliste als „nichtVorhanden“ markiert
Vorbedingungen	In der Produktliste, Rezeptproduktliste und Getränke-Tabelle sind Produkte vorhanden, die in die Einkaufsliste hinzugefügt werden müssen.
Eingehende Informationen	-
Ergebnisse	Die Einkaufsliste ist gefüllt.
Nachbedingungen	<ul style="list-style-type: none"> • Die Produkte in der Produktliste wurden als „ausgewaehlt“ markiert. • Die Rezeptproduktliste ist geleert worden. • In der Getränke-Tabelle wurden die ausgewählten Produkte als „ausgewaehlt“ markiert.
Ablauf	<ol style="list-style-type: none"> 1. Produktliste bearbeiten Vergleichen der Produkte in der Einkaufsliste mit denen in der Produktliste. Nicht vorhandene werden der Einkaufsliste hinzugefügt und die Bemerkung wird auf „ausgewaehlt“ geändert. 2. Rezeptproduktliste bearbeiten Vergleichen der Produkte in der Einkaufsliste mit denen in der Rezeptproduktliste. Bei vorhandenen Produkten wird die Menge addiert. Nicht vorhandene werden der Einkaufsliste hinzugefügt.

	3. Getränke-Tabelle bearbeiten Vergleichen der Produkte in der Einkaufsliste mit denen in der Getränke-Tabelle. Vorhandene Produkte werden nicht hinzugefügt. Nicht vorhandene werden der Einkaufsliste hinzugefügt.
Risiko	Hohes Risiko
Verbindlichkeit, Priorität	Hohe Priorität, unverzichtbar

Tabelle 8: Use Case 4 Einkaufsliste erstellen

Der Use Case Einkaufsliste erstellen beinhaltet die Abarbeitung von drei Tabellen. Damit die Übersicht erhalten bleibt, wird von jedem Schritt je ein Systemablaufmodell erstellt.

Die unten aufgeführten Abbildungen stellen folgende Schritte bei der Erstellung der Einkaufsliste dar:

- Produktliste bearbeiten
- Rezeptproduktliste bearbeiten
- Getränke-Tabelle bearbeiten

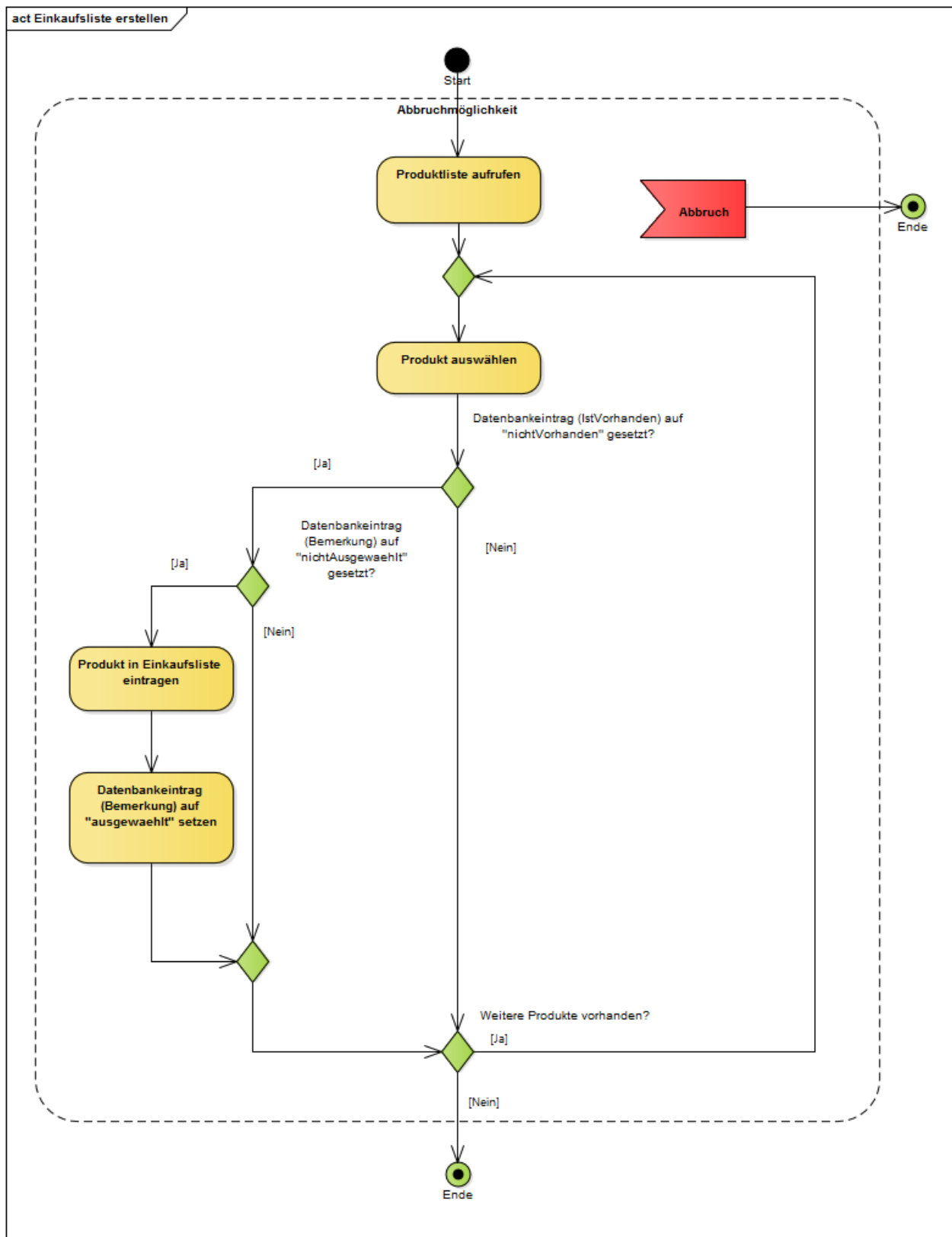


Abbildung 27: Systemablaufmodell Einkaufsliste erstellen – Produktliste

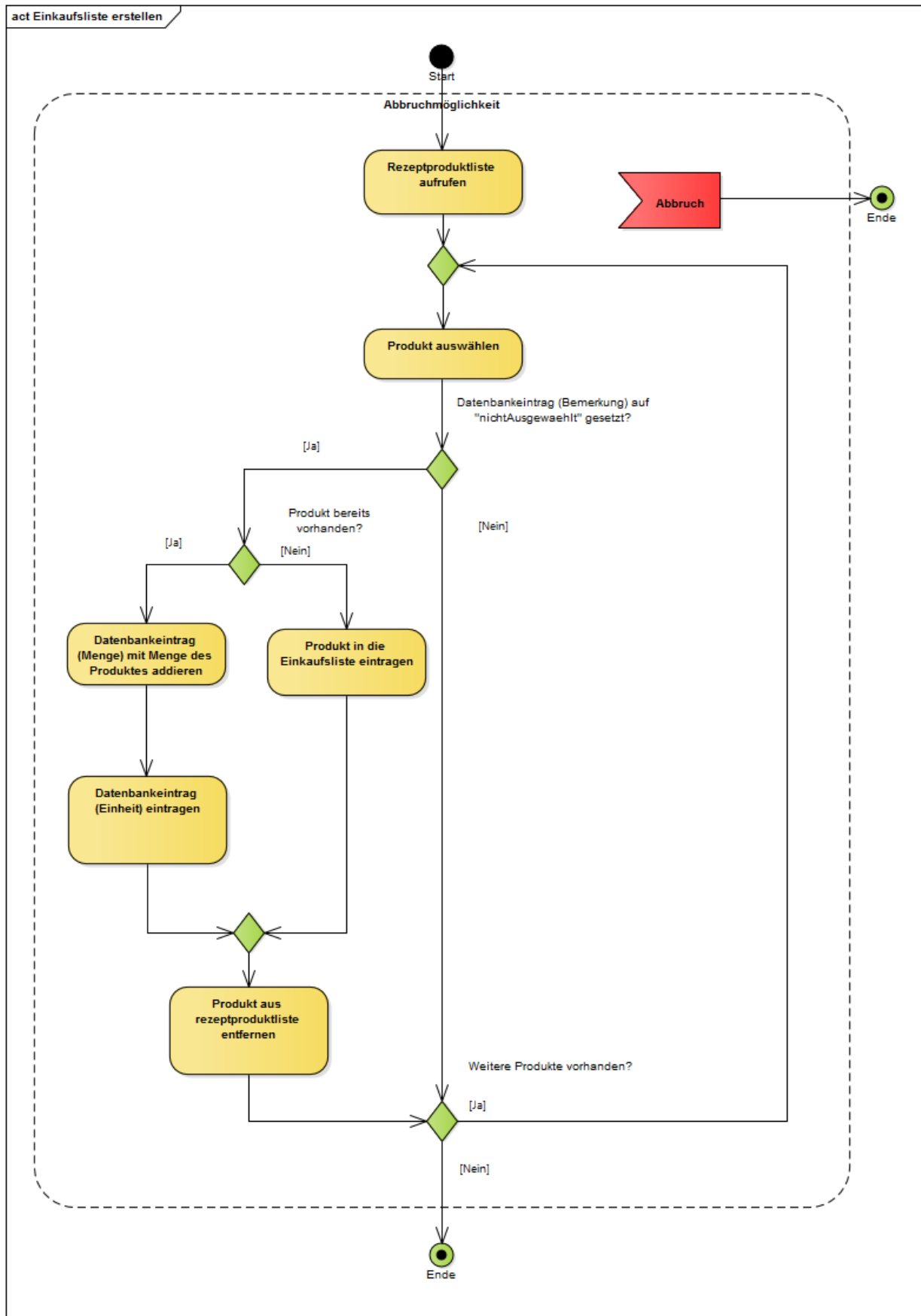


Abbildung 28: Systemablaufmodell Einkaufsliste erstellen – Rezeptproduktliste

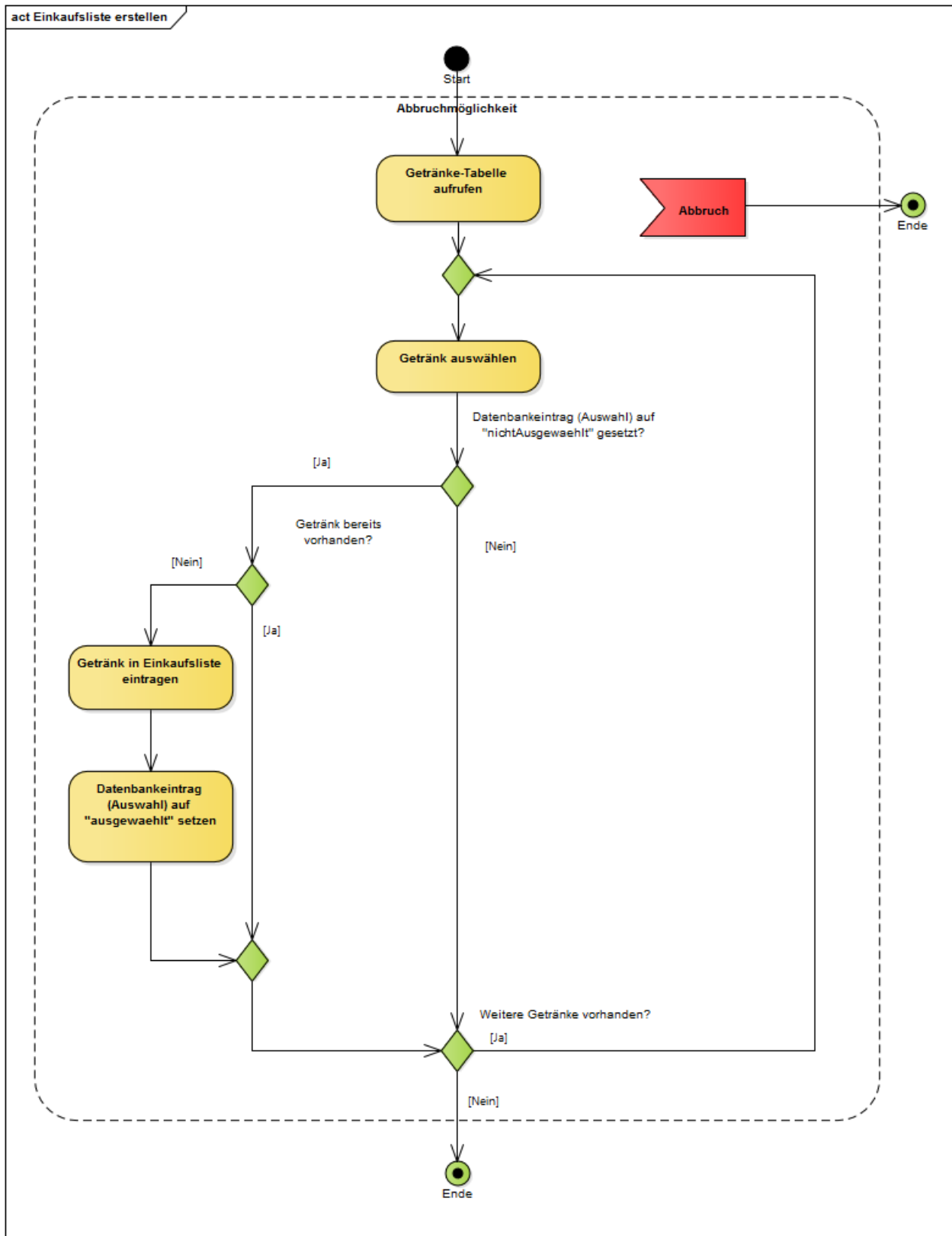


Abbildung 29: Systemablaufmodell Einkaufsliste erstellen – Getränke

3.1.6.5 Use Case Einkaufsliste aktualisieren

Name	Einkaufsliste aktualisieren
Kurzbeschreibung	<p>Das Programm ruft die Klasse, in der die Einkaufsliste aktualisiert wird. Dabei kontrolliert das Programm, ob ein Produkt in die Datenbank als „vorhanden“ eingetragen wurde.</p> <p>Ist das der Fall, wird das Produkt aus der Einkaufsliste entfernt.</p>
Akteure	-
Auslöser	Das Programm startet die Einkaufslistenaktualisierung.
Vorbedingungen	-
Eingehende Informationen	-
Ergebnisse	Produkte, die in der Produktliste vorhanden sind, werden aus der Einkaufsliste entfernt.
Nachbedingungen	Der Benutzer erhält eine aktuelle Einkaufsliste.
Ablauf	<ol style="list-style-type: none"> 1. Produkt suchen Die Anwendung sucht in der Produktliste nach dem ausgewählten Produkt aus der Einkaufsliste. 2. Produkt gefunden Ist in der Produktliste ein Produkt vorhanden und ist als „vorhanden“ markiert, wird dieses Produkt aus der Einkaufsliste entfernt. 3. Produkt nicht gefunden Wurde kein Produkt gefunden oder das ausgewählte Produkt als „nichtVorhanden“ markiert, wird das Produkt aus der Einkaufsliste nicht gelöscht.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, unverzichtbar

Tabelle 9: Use Case 5 Einkaufsliste aktualisieren

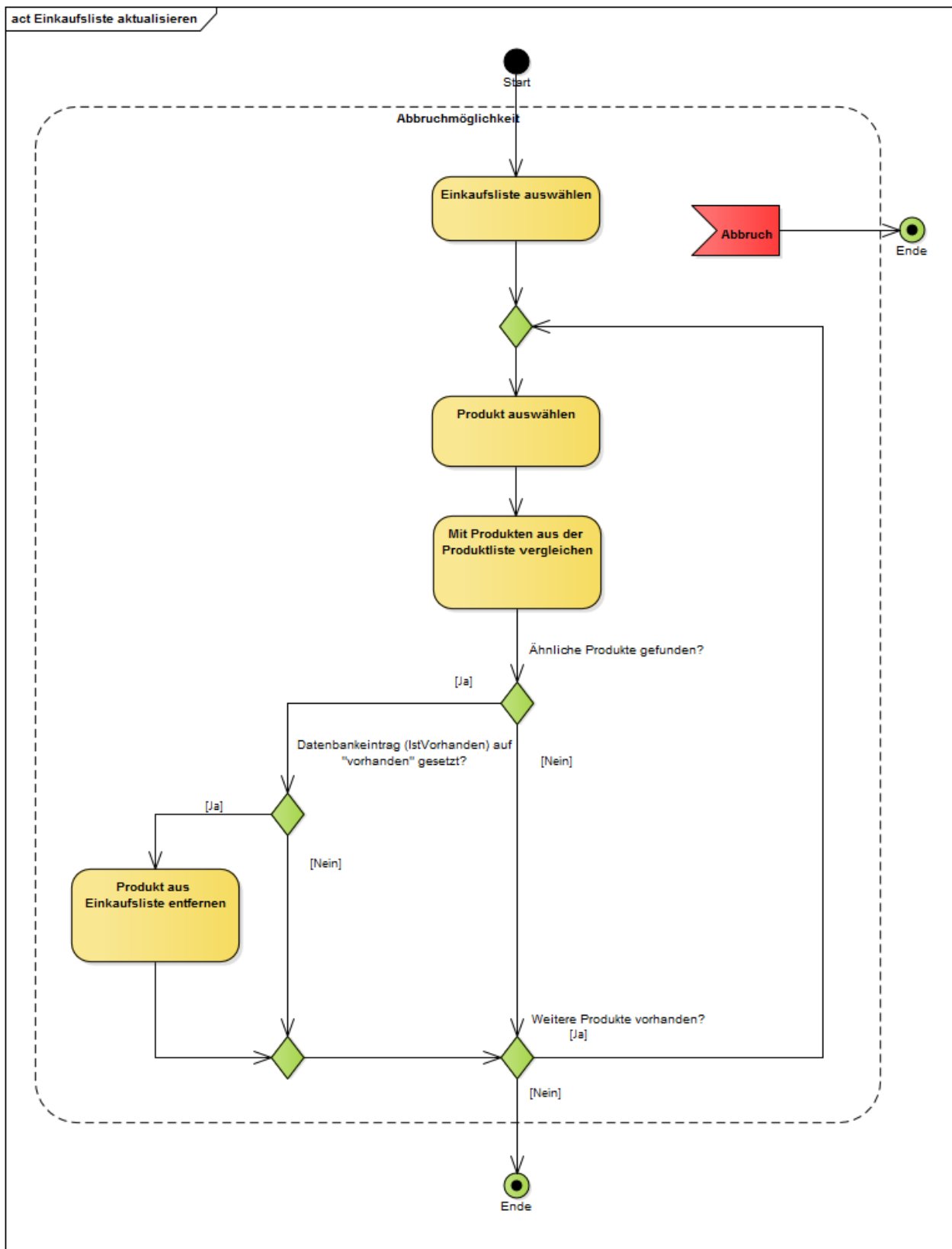


Abbildung 30: Systemablaufmodell Einkaufsliste aktualisieren

3.1.6.6 Use Case Foto erstellen

Name	Foto erstellen
Kurzbeschreibung	Die JaRasPi-Anwendung erstellt in bestimmten Abständen ein Foto vom Inhalt eines Lebensmittelschranks oder einer Fruchtschale und speichert es auf dem Raspberry Pi.
Akteure	-
Auslöser	Eingestellter Zeitpunkt
Vorbedingungen	-
Eingehende Informationen	-
Ergebnisse	Ein Foto vom Inhalt des Lebensmittelschranks oder einer Fruchtschale wurde erstellt.
Nachbedingungen	-
Ablauf	<ol style="list-style-type: none"> 1. Kamera auswählen Einmal in drei Stunden wird in der JaRasPi-Anwendung die KameraKlasse gestartet und die Kamera des Raspberry Pi ausgewählt. 2. Foto erstellen Als nächstes wird ein neues Bild erstellt und das alte überschrieben.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, verzichtbar

Tabelle 10: Use Case 6 Foto erstellen

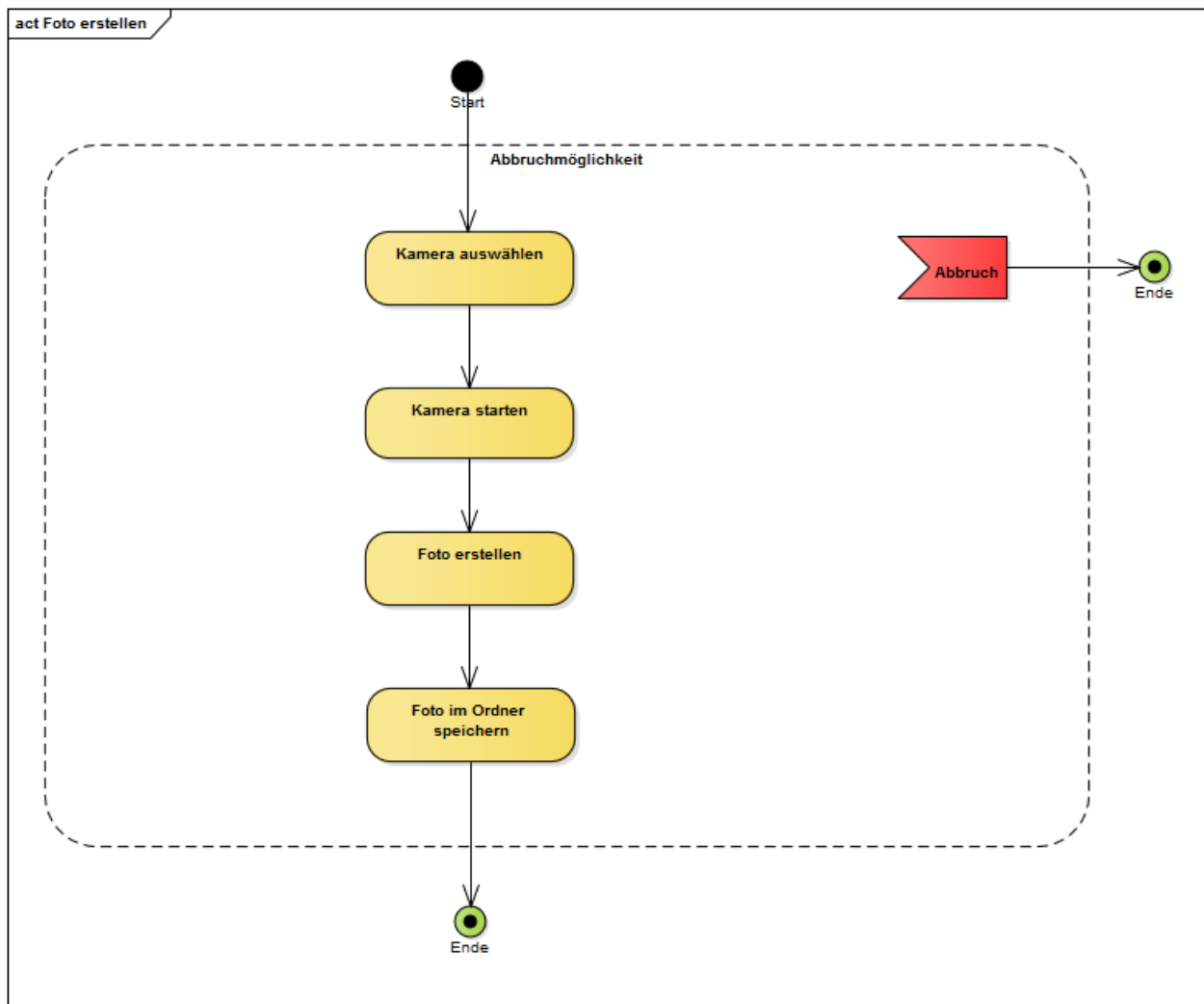


Abbildung 31: Systemablaufmodell Foto erstellen

3.1.6.7 Use Case Einkaufsliste auswählen

Name	Einkaufsliste auswählen
Kurzbeschreibung	<p>Der Benutzer wählt auf der Website „Einkaufsliste anzeigen“ aus. Der Webserver greift mit einem lesenden Zugriff auf die Datenbank zu.</p> <p>Nachdem die Daten ausgelesen wurden, werden diese dem Benutzer in Form einer digitalen Einkaufsliste bereitgestellt.</p>
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer möchte die Einkaufsliste ansehen. Dazu ruft er mit einem internetfähigen Gerät z.B. Smartphone, die Website auf und drückt auf „Einkaufsliste anzeigen“.
Vorbedingungen	-
Eingehende Informationen	„Einkaufsliste anzeigen“ wurde gedrückt.
Ergebnisse	Der Webserver liest die Einkaufsliste-Tabelle und erstellt eine digitale Einkaufsliste.
Nachbedingungen	Dem Benutzer wird eine digitale Einkaufsliste angezeigt.
Ablauf	<ol style="list-style-type: none"> 1. Einkaufsliste anzeigen Der Benutzer ruft den Webserver auf und drückt auf „Einkaufsliste anzeigen“. 2. Auf die Einkaufsliste zugreifen Der Webserver greift auf die Einkaufsliste mit einem lesenden Zugriff zu und erstellt die Einkaufsliste.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, unverzichtbar

Tabelle 11: Use Case 7 Einkaufsliste auswählen

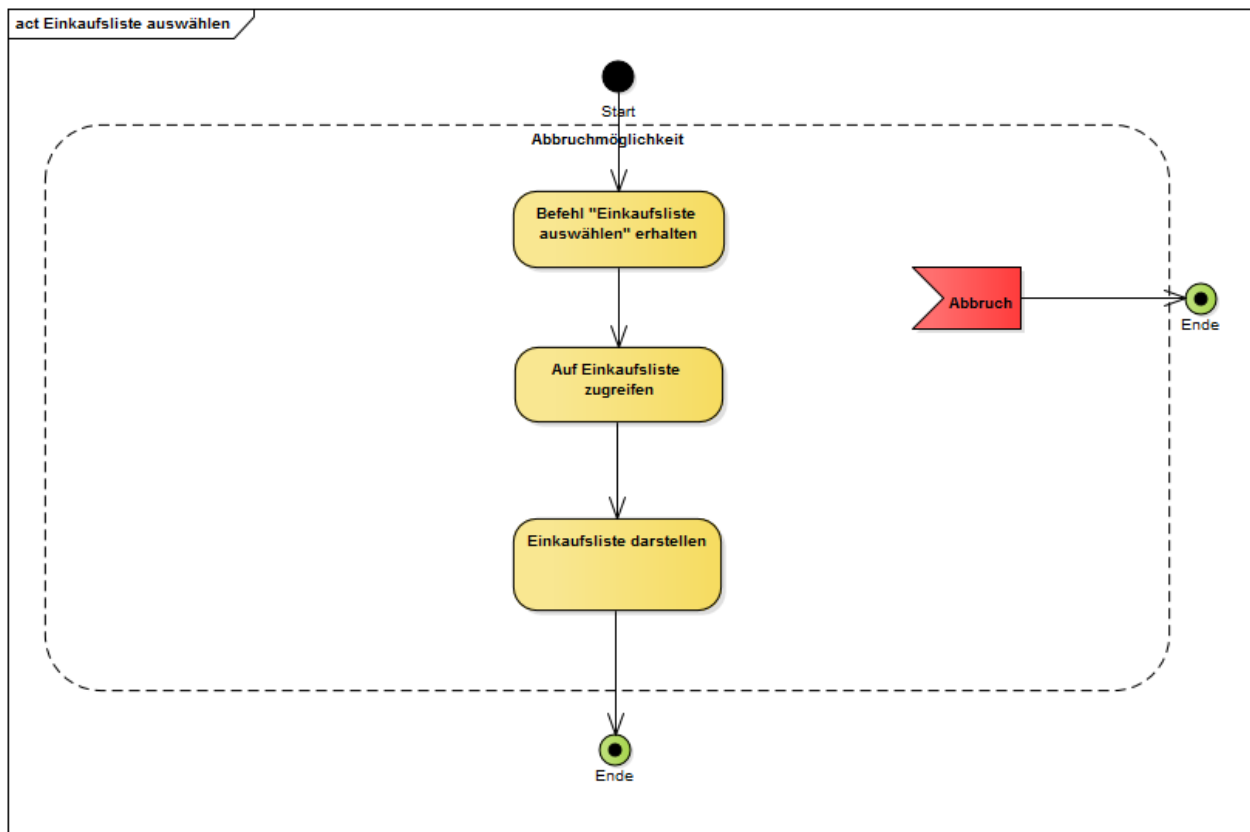


Abbildung 32: Systemablaufmodell Einkaufsliste auswählen

3.1.6.8 Use Case Inhalt anzeigen

Name	Inhalt anzeigen
Kurzbeschreibung	<p>Der Benutzer wählt die Option „Inhalt anzeigen“ aus.</p> <p>Der Webserver greift auf den Ordner mit den erstellten Fotos in dem Raspberry Pi zu und stellt das Foto dem Benutzer bereit.</p>
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer möchte den Inhalt der Fruchtschale oder eines Speiseschranks ansehen.
Vorbedingungen	-
Eingehende Informationen	„Inhalt anzeigen“ wurde gedrückt.
Ergebnisse	Der Webserver stellt das zuvor erstellte Bild auf der Webseite bereit.
Nachbedingungen	Dem Benutzer wird ein Foto vom erstellten Inhalt angezeigt.
Ablauf	<ol style="list-style-type: none"> 1. Inhalt anzeigen Der Benutzer ruft den Webserver auf und drückt auf „Inhalt anzeigen“. 2. Ordner öffnen Der Webserver erkennt den Befehl und greift auf den Ordner, in dem das Bild gespeichert ist, zu. 3. Bild darstellen Der Webserver stellt das Bild dem Benutzer zur Verfügung.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, unverzichtbar

Tabelle 12: Use Case 8 Inhalt anzeigen

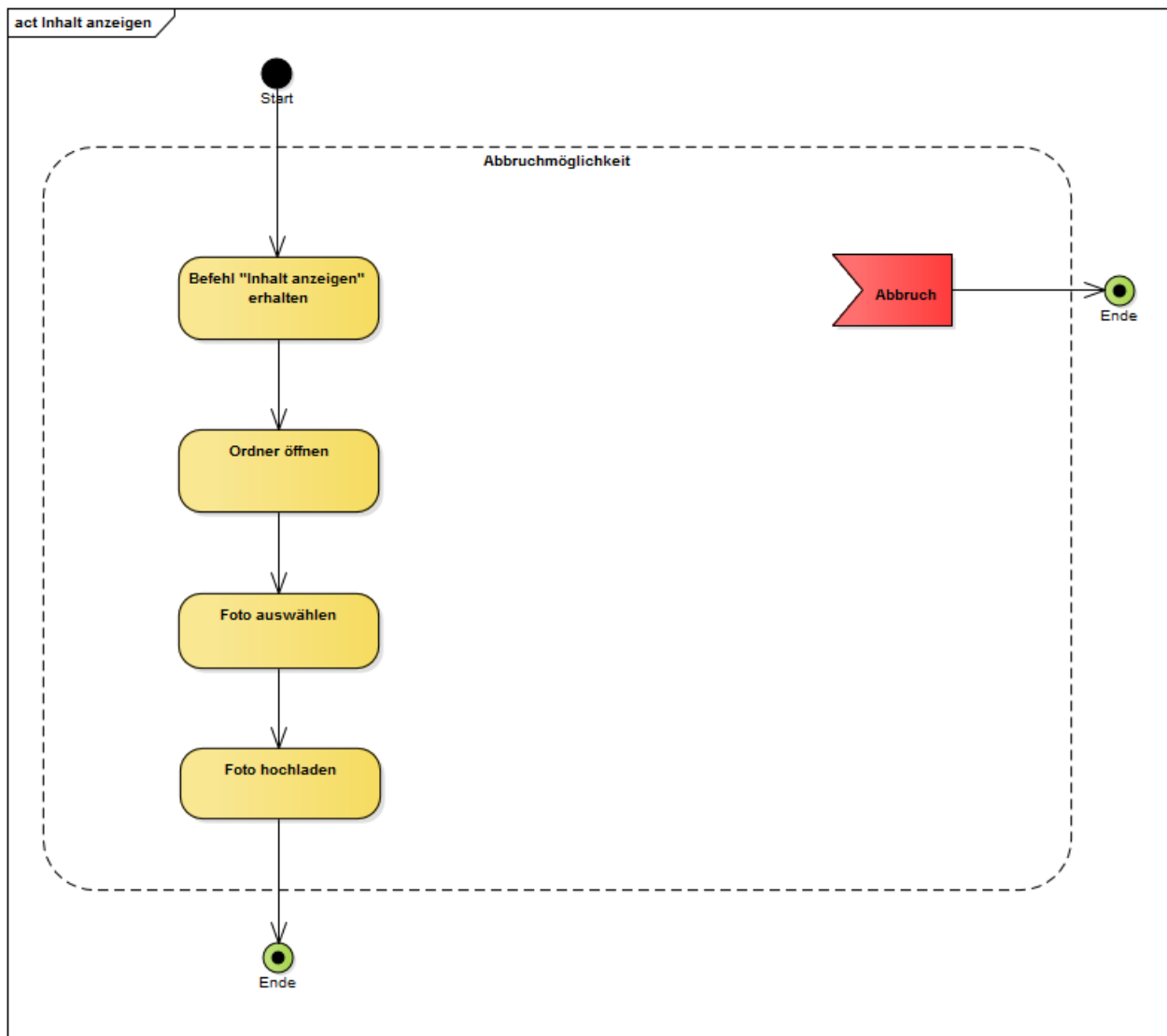


Abbildung 33: Systemablaufmodell Inhalt anzeigen

3.1.6.9 Use Case Rezept auswählen

Name	Rezept auswählen
Kurzbeschreibung	Der Benutzer wählt auf der Website ein Rezept aus. Der Webserver liest die benötigten Daten aus und markiert in der Rezept-Tabelle das ausgewählte Rezept.
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer wählt ein Rezept aus.
Vorbedingungen	-
Eingehende Informationen	Ein Rezept wurde ausgewählt.
Ergebnisse	Das Rezept wurde in der Rezept-Tabelle als „ausgewählt“ markiert.
Nachbedingungen	Dem Benutzer wird mitgeteilt, dass ein Rezept ausgewählt wurde.
Ablauf	<ol style="list-style-type: none"> 1. Rezept auswählen Der Benutzer wählt ein Rezept aus. 2. Rezept-Tabelle aktualisieren Der Webserver erkennt das Rezept und ändert in der Rezept-Tabelle, die Bemerkung auf „ausgewählt“ um. 3. Auswahl bestätigen Nach dem Eintragen wird der Benutzer über die erfolgreiche Aktualisierung informiert.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, unverzichtbar

Tabelle 13: Use Case 9 Rezept auswählen

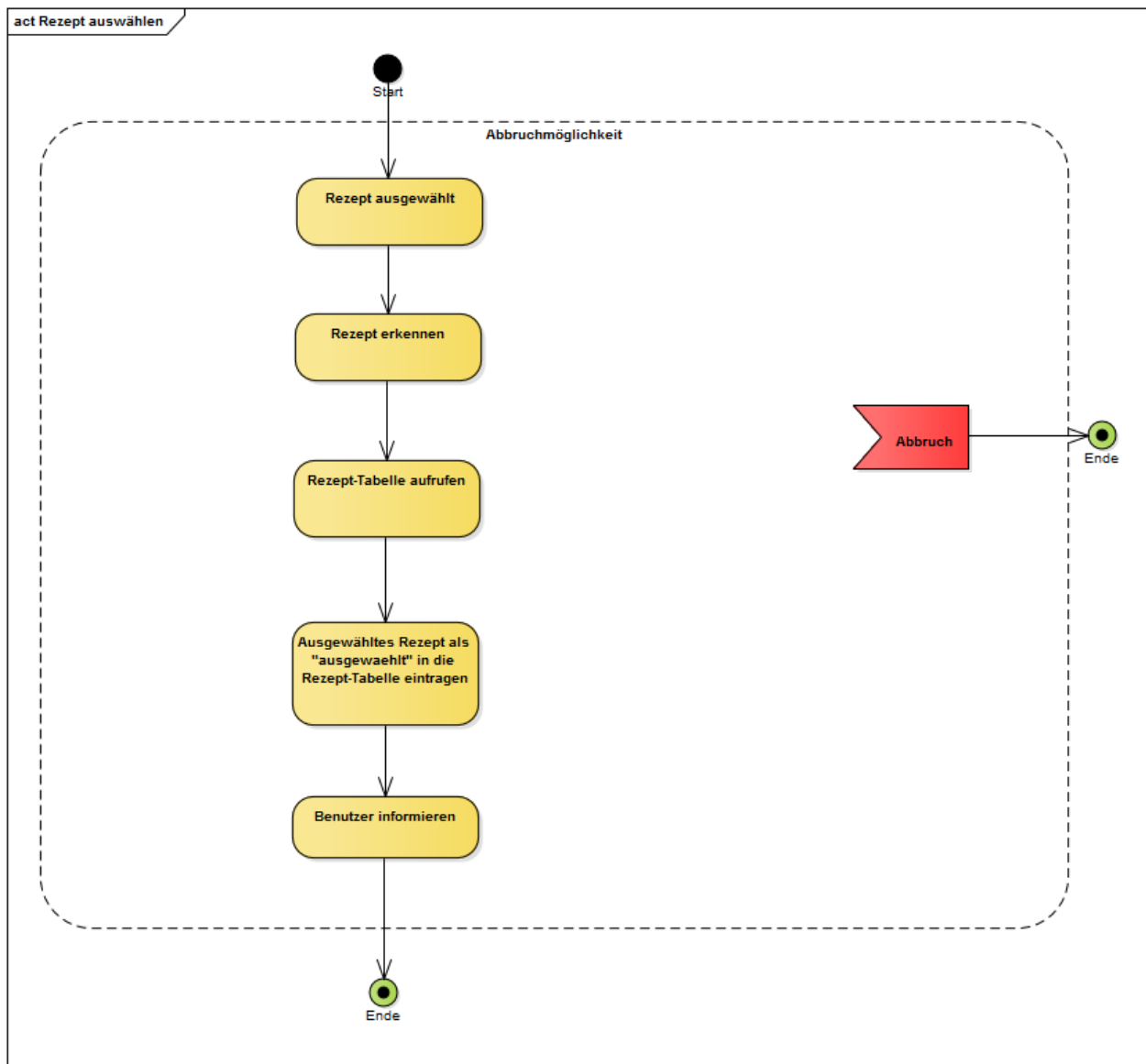


Abbildung 34: Systemablaufmodell Rezept auswählen

3.1.6.10 Use Case Rezept erstellen

Name	Rezept erstellen
Kurzbeschreibung	Der Benutzer wählt auf der Website „Rezept erstellen“ aus. Es erscheinen mehrere Eingabefelder, in die der Benutzer die benötigten Daten einträgt und das Rezept erstellt.
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer erstellt ein neues Rezept.
Vorbedingungen	-
Eingehende Informationen	„Rezept erstellen“ wurde gedrückt.
Ergebnisse	Der Webserver liest die eingegebenen Daten und erstellt ein neues Rezept.
Nachbedingungen	Der Benutzer wird informiert, dass das Rezept erfolgreich erstellt wurde.
Ablauf	<ol style="list-style-type: none"> 1. Befehl erhalten Der Benutzer fügt alle benötigten Daten in das Rezept ein und wählt „Rezept erstellen“ aus. 2. Daten erhalten Der Webserver erhält die Daten und erstellt anhand dieser das Rezept. 3. In Rezept-Tabelle eintragen Der Webserver trägt das Rezept in die Rezept-Tabelle ein. 4. Benutzer informieren Als letztes informiert der Webserver den Benutzer, dass das Rezept erstellt wurde.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, unverzichtbar

Tabelle 14: Use Case 10 Rezept erstellen

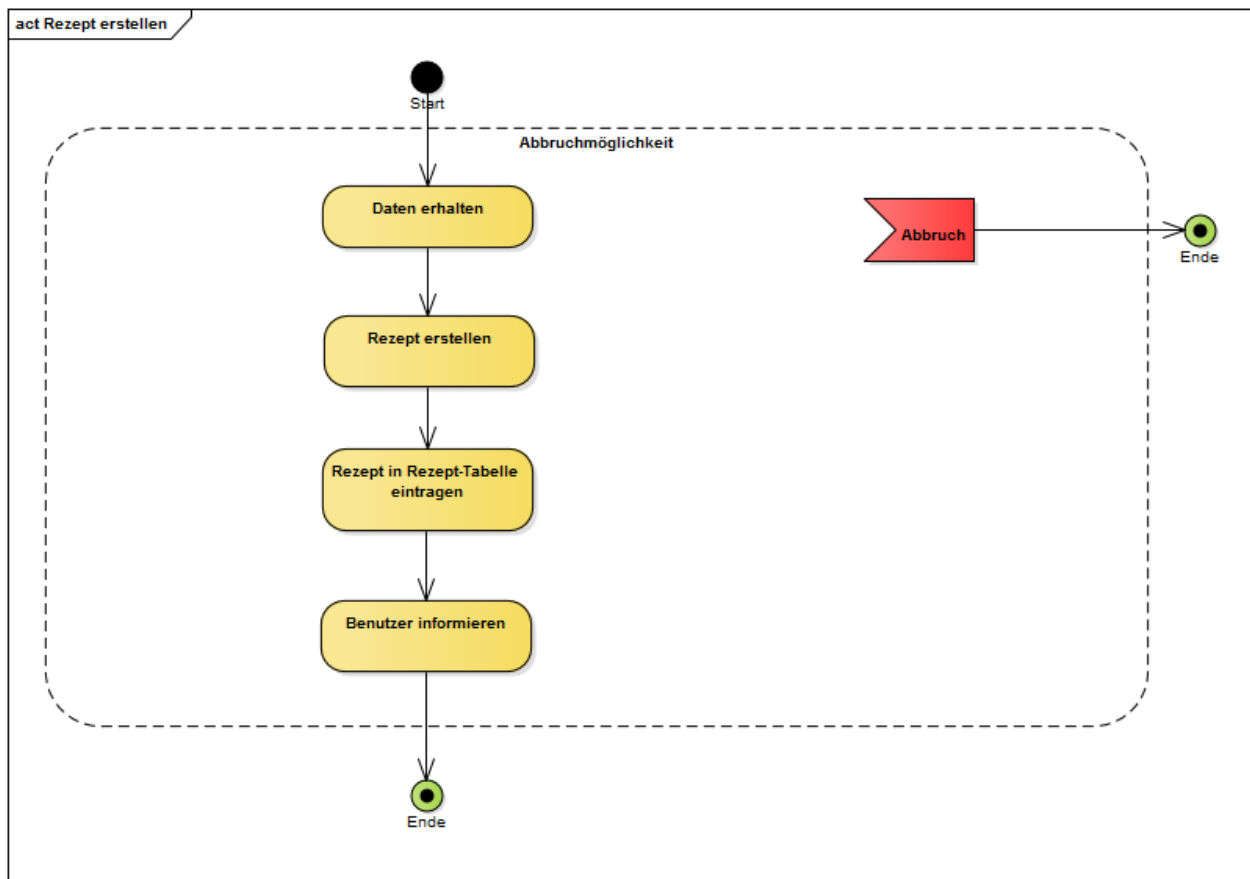


Abbildung 35: Systemablaufmodell Rezept erstellen

3.1.6.11 Use Case Produkt entfernen

Name	Produkt entfernen
Kurzbeschreibung	Wurde „Produkt entfernen“ ausgewählt, kann der Benutzer manuell Produkte aus der Einkaufsliste entfernen. Wurde ein Produkt entfernt, wird der Benutzer darüber informiert, dass das Austragen erfolgreich gewesen ist.
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer möchte ein Produkt aus der Einkaufsliste entfernen.
Vorbedingungen	Produkte müssen in der Einkaufsliste vorhanden sein.
Eingehende Informationen	Das zu löschende Produkt.
Ergebnisse	Das Produkt wurde aus der Einkaufsliste entfernt.
Nachbedingungen	Die aktualisierte Einkaufsliste wird angezeigt.
Ablauf	<ol style="list-style-type: none"> 1. Produkt auswählen Der Benutzer wählt das zu entfernende Produkt aus. 2. Auf die Einkaufsliste zugreifen Der Webserver erkennt das Produkt und entfernt es aus der Einkaufsliste. 3. Einkaufsliste bereitstellen Der Webserver stellt die aktualisierte Einkaufsliste dem Benutzer zur Verfügung.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, unverzichtbar

Tabelle 15: Use Case 11 Produkt entfernen

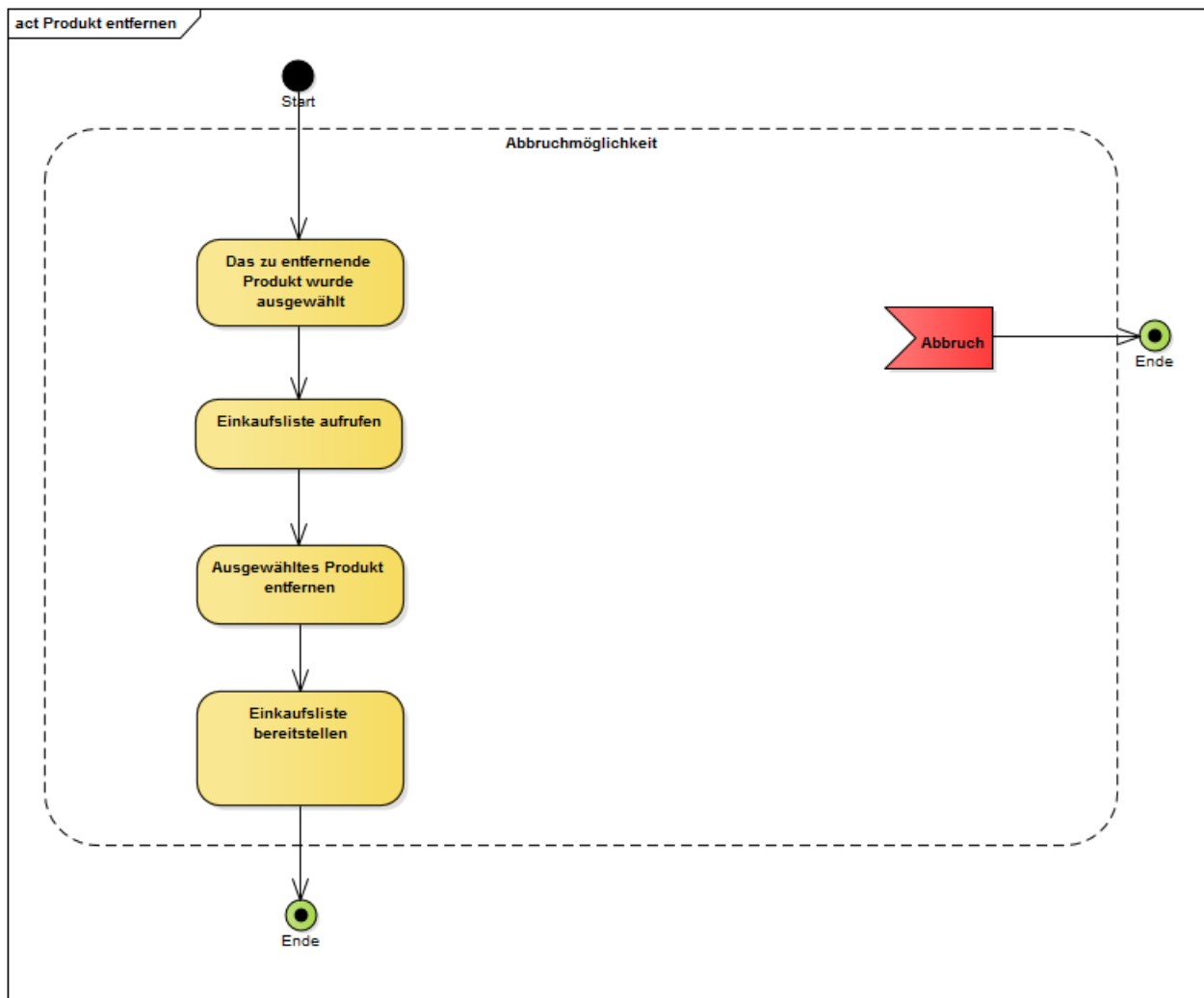


Abbildung 36: Systemablaufmodell Produkt entfernen

3.1.6.12 Use Case Produkt hinzufügen

Name	Produkt hinzufügen
Kurzbeschreibung	Wurde „Produkt hinzufügen“ ausgewählt, kann der Benutzer manuell Produkte in die Einkaufsliste hinzufügen. Wurde ein Produkt hinzugefügt, wird der Benutzer darüber informiert, dass das Eintragen erfolgreich gewesen ist.
Akteure	Benutzer(Verbraucher)
Auslöser	Der Benutzer möchte ein Produkt in die Einkaufsliste hinzufügen.
Vorbedingungen	-
Eingehende Informationen	Das hinzufügende Produkt.
Ergebnisse	Das Produkt wurde in die Einkaufsliste hinzugefügt.
Nachbedingungen	Die aktualisierte Einkaufsliste wird angezeigt.
Ablauf	<ol style="list-style-type: none"> 1. Produkt auswählen Der Benutzer fügt das Produkt hinzu. 2. Auf die Einkaufsliste zugreifen Der Webserver erkennt die Daten und fügt sie der Einkaufsliste hinzu. 3. Einkaufsliste bereitstellen Der Webserver stellt die aktualisierte Einkaufsliste dem Benutzer zur Verfügung. 4. Benutzer informieren Der Benutzer wird informiert, dass das Produkt erfolgreich in die Einkaufsliste aufgenommen wurde.
Risiko	Mittleres Risiko
Verbindlichkeit, Priorität	Mittlere Priorität, unverzichtbar

Tabelle 16: : Use Case 12 Produkt hinzufügen

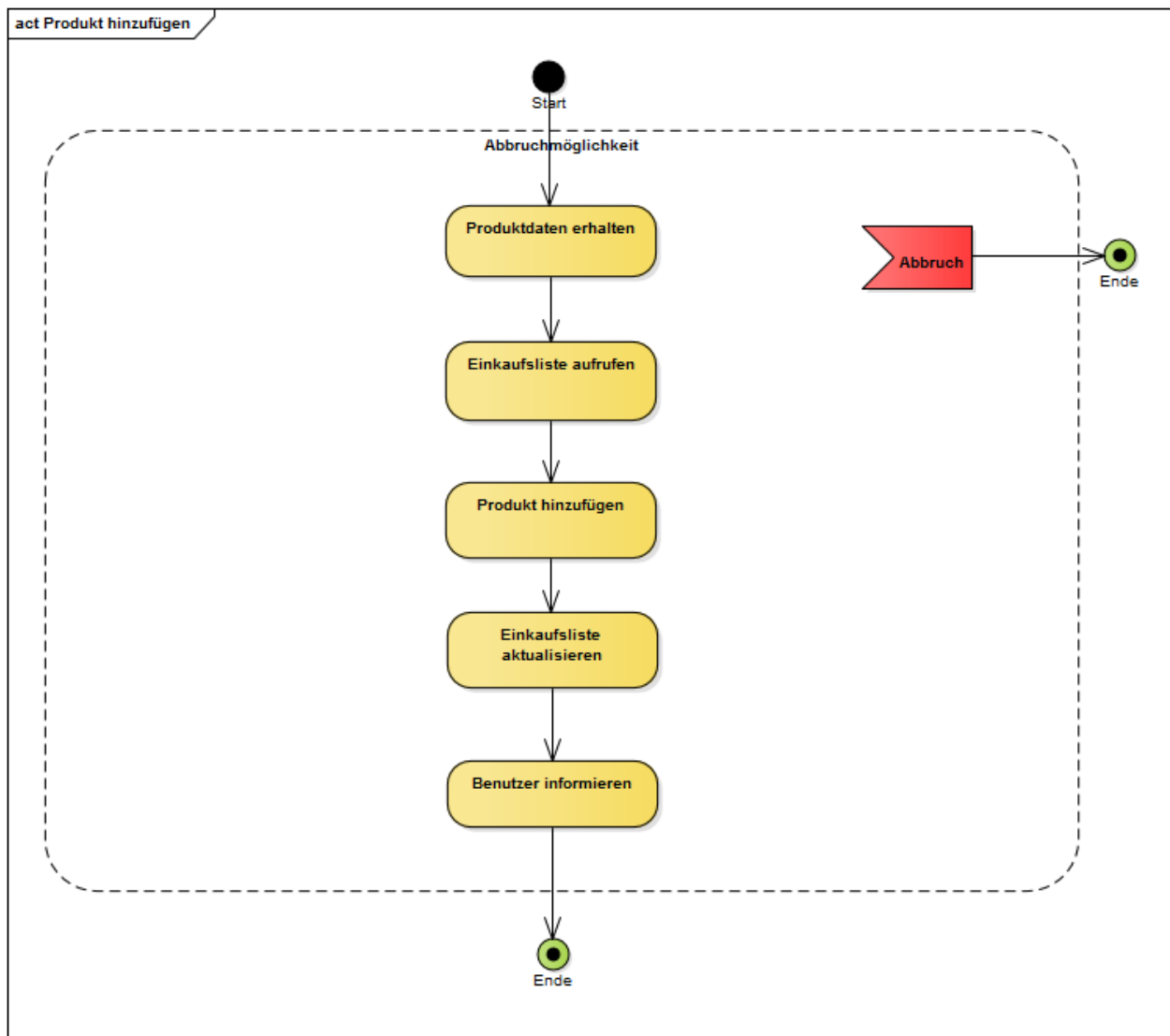


Abbildung 37: Systemablaufmodell Produkt hinzufügen

3.1.7 Beziehungen zwischen Anwendungsfällen

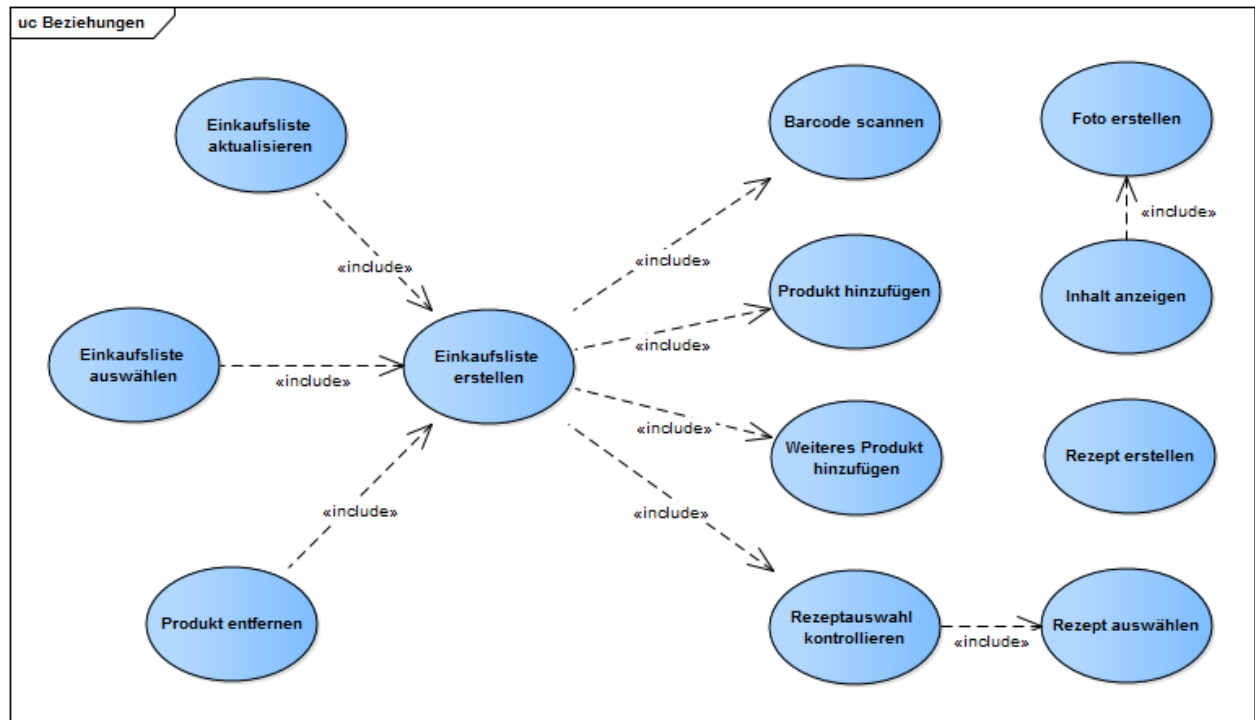


Abbildung 38: Beziehungen zwischen den Anwendungsfällen

3.1.8 Nichtfunktionale Anforderungen

3.1.8.1 Look and Feel

Der Benutzer kann sowohl auf dem Smartphone, als auch am Computer die Website aufrufen. Wichtig ist jedoch, dass die Ansicht für das Smartphone so angepasst wird, dass der Benutzer die komplette Übersicht beibehält. Dies wird erreicht, indem Responsive Webdesign verwendet wird. Alle Funktionen, die der Benutzer auf dem Desktop Computer zu Verfügung gestellt bekommt, werden auch auf dem Smartphone vorhanden sein.

Während die Anwendung und der Webserver auf einem Raspberry Pi ausgeführt werden müssen, kann der Aufruf der Webseite unabhängig von dem System erfolgen.

3.1.8.2 Usability

3.1.8.2.1 Benutzerfreundlichkeit

JaRasPi-Anwendung

Die JaRasPi-Anwendung ist an den Benutzer gerichtet. Nach dem Start der Anwendung erscheint ein GUI, wo der Benutzer die Anwendung starten oder anhalten kann. Das Starten oder Anhalten findet über zwei Buttons statt, die je nachdem, welcher Button gedrückt wurde, ihre Farben wechseln. Wurde der Start-Button gedrückt, wechselt dieser in die Farbe Rot und der Anhalten-Button, wechselt die Farbe in Grün um. Wird der Anhalten-Button gedrückt, geschieht das Gegenteil.

Da der Benutzer ausversehen zweimal die Anwendung starten könnte, wurde eine Sicherung eingebaut, damit nur der erste Start funktioniert und die weiteren ignoriert werden.

Nach dem Start kann der Barcode eingescannt werden. Nach dem Einscannen wird dieser kurz im Textfeld dargestellt und nach dem Eintragen wieder entfernt. So wird dem Benutzer mitgeteilt, dass der Barcode in die Datenbank aufgenommen wurde.

Es gibt einen weiteren Button, den der Benutzer betätigen kann, wenn er das Produkt in die Datenbank buchen möchte. Scant er das Produkt ohne den Button zu betätigen, wird die Anzahl des Produktes um die Menge 1 verringert. Wird der Button betätigt wird die Anzahl um 1 erhöht.

JaRasPi-Website

Während das Anzeigen der Einkaufsliste oder des Inhaltes der Fruchtschale durch einen einfachen Klick auf den jeweilige Button erledigt ist, soll das Auswählen von Rezepten, Hinzufügen in die Einkaufsliste oder das Erstellen von neuen Rezepten, den Benutzer informieren, dass Ihre Aktion erfolgreich gewesen ist.

Weiterhin wird der Benutzer beim Eintragen von Produkten in die Einkaufsliste informiert, welche Felder wichtig sind und welche ausgelassen werden können.

Durch den Farbwechsel der Buttons wird der Benutzer informiert, auf welchem Navigationspunkt er sich momentan befindet.

3.1.8.2.2 Erlernbarkeit

JaRasPi-Anwendung

Das Verwenden der JaRasPi-Anwendung bedarf einer kurzen Einweisung durch den Programmierer. Das Starten der Anwendung und das Abscannen der Produkte soll unerfahrenen Benutzern erklärt werden.

JaRasPi-Website

Die Webseite ist intuitiv aufgebaut. Durch aussagekräftige Bezeichnungen der Buttons und Überschriften wird der Benutzer durch die Webseite begleitet. Durch Farbveränderungen wird der Benutzer informiert, auf welchem Navigationspunkt er sich momentan befindet. Neben dem Benutzerhandbuch findet eine Erläuterung aller Funktionen statt.

3.2 Systementwurf

Nachdem im Kapitel 3.1 die pflichtenheftspezifischen Punkte abgearbeitet wurden, wird im Kapitel 3.2 der Systementwurf vorgestellt.

3.2.1 Subsystemdiagramm

Das Subsystemdiagramm wird in drei Bereiche unterteilt. Und zwar in den JaRasPi-Webserver, MySQL-Datenbank und der JaRasPi-Anwendung.

Wichtig zu erwähnen ist, dass bei dem Webserver und der Datenbank nur die Haupt-Webseiten/-Tabellen dargestellt wurden.

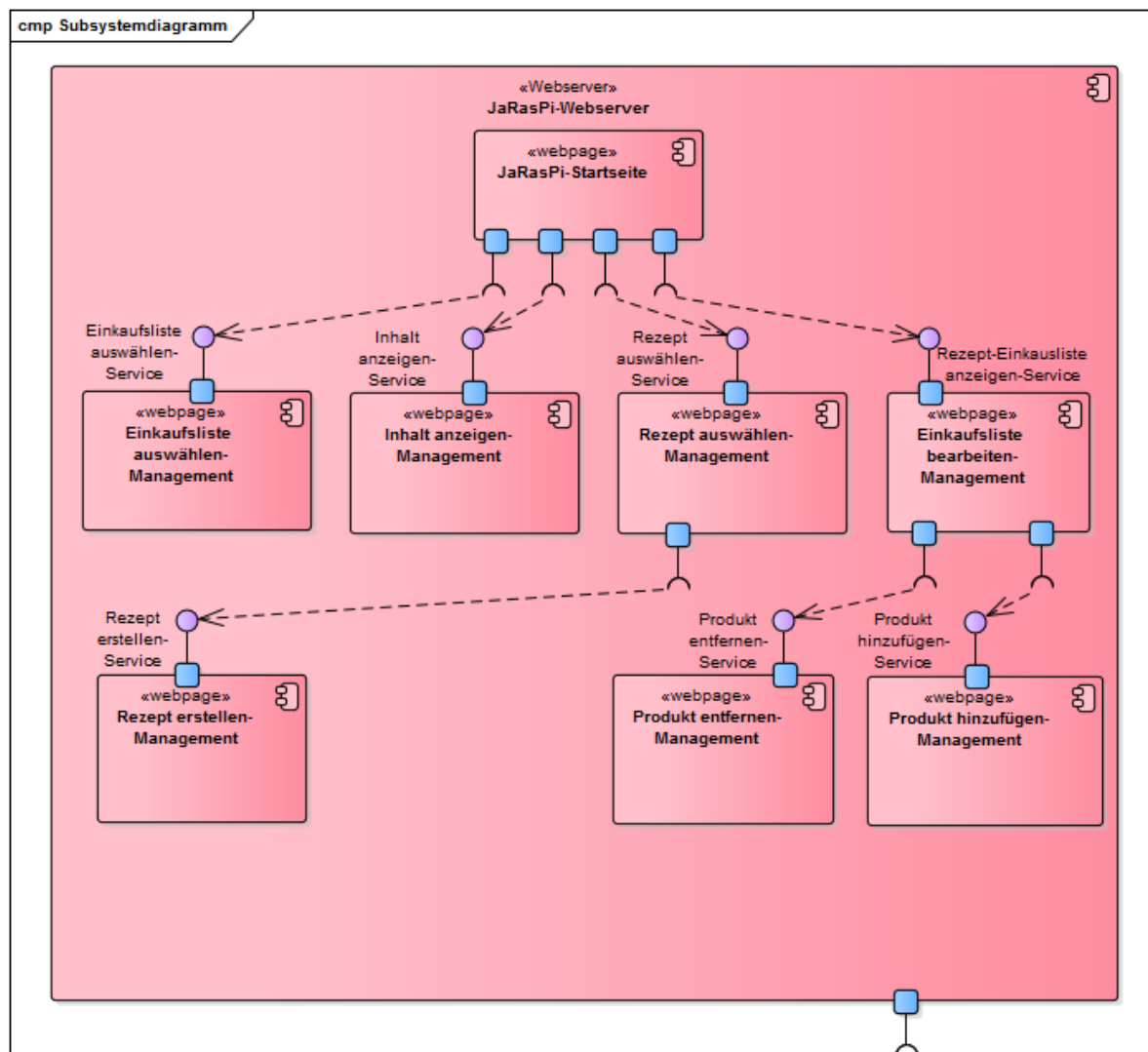


Abbildung 39: Subsystemdiagramm Nr.1 – JaRasPi-Webserver

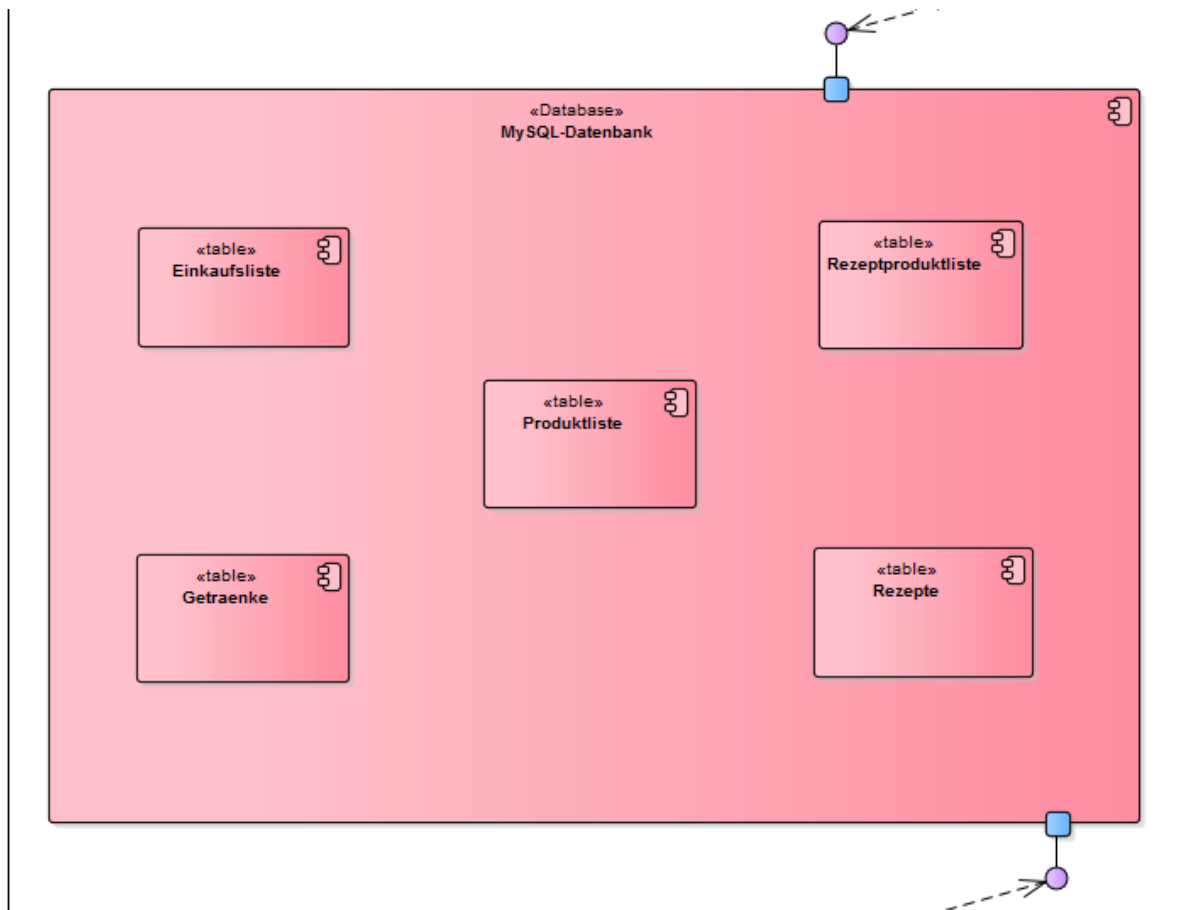


Abbildung 40: Subsystemdiagramm Nr.2 – MySQL-Datenbank

Die Datenbank verfügt über insgesamt 18 Tabellen. Die wichtigsten wurden in der Abbildung 39 dargestellt.

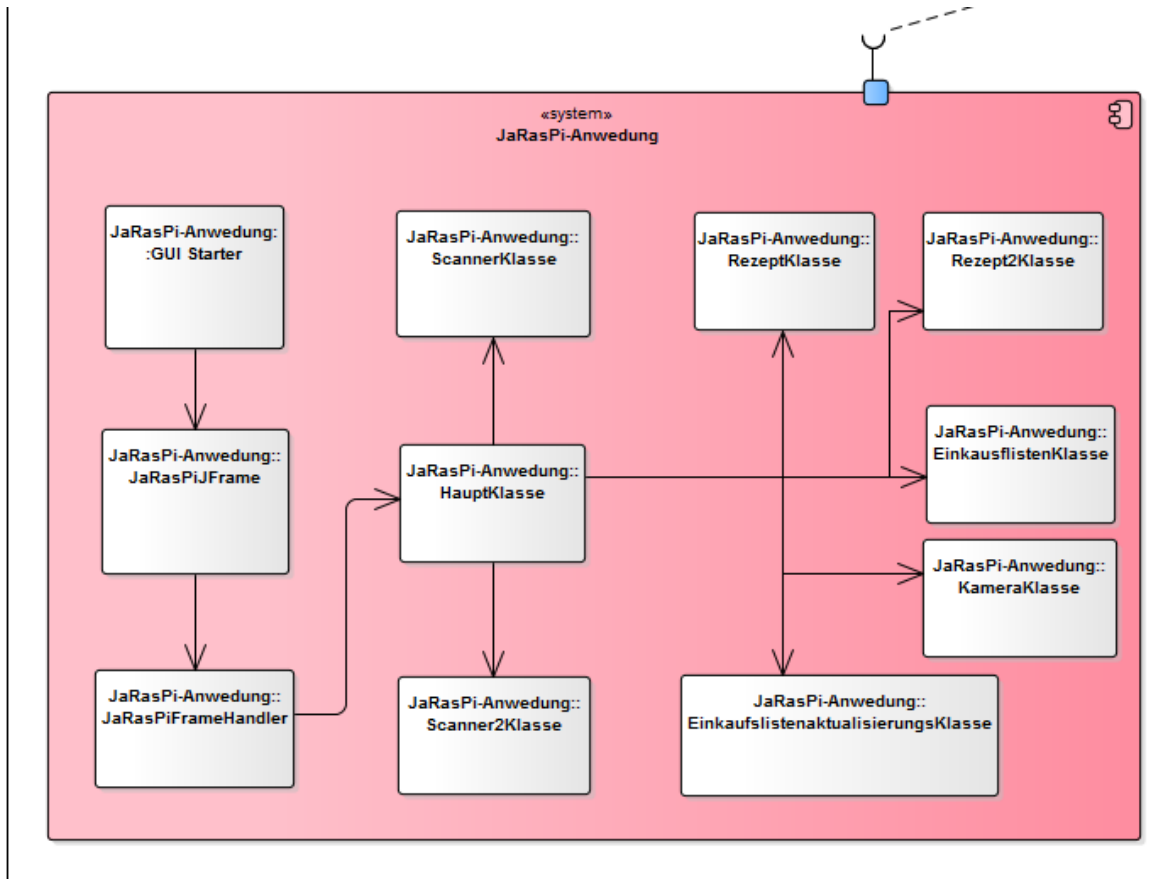


Abbildung 41: Subsystemdiagramm Nr.3 - JaRasPi-Anwendung

3.2.2 Verteilungsdiagramm

Der nächste Punkt stellt die Verteilung des Systems dar. Dabei wird gezeigt, welche Komponenten auf welcher Hardware laufen, wie sie konfiguriert sind und wie die Kommunikationsbeziehungen stattfinden. Da alle Komponenten dieses System auf dem Raspberry Pi laufen werden, sieht das Verteilungsdiagramm wie in Abbildung 41 dargestellt aus.⁵⁶

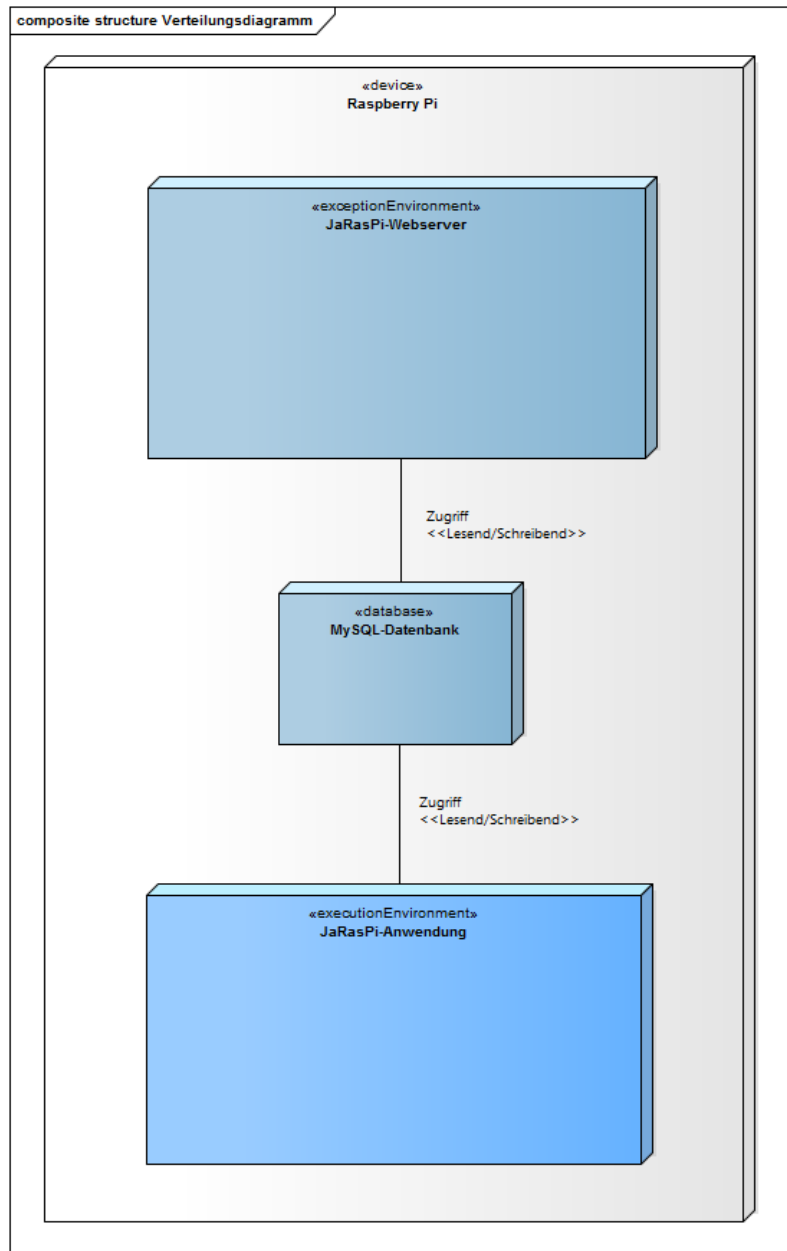


Abbildung 42: Verteilungsdiagramm

⁵⁶ Vgl. (Oestereich, 2009), S. 332 – 333.

3.2.3 Benutzeroberfläche Webserver

Die nächsten Abbildungen zeigen wie die JaRasPi-Webseite aufgebaut wird. Hierbei handelt es sich lediglich um Zeichnungen. Die finale Version wird sich von den Abbildungen unterscheiden.



Abbildung 43: Startseite der Webseite



Abbildung 44: Einkaufsliste auswählen



Abbildung 45: Inhalt anzeigen



Abbildung 46: Rezept auswählen



Abbildung 47: Rezept-Einkaufsliste anzeigen

3.2.4 Benutzeroberfläche JaRasPi



Abbildung 48: JaRasPi Startoberfläche

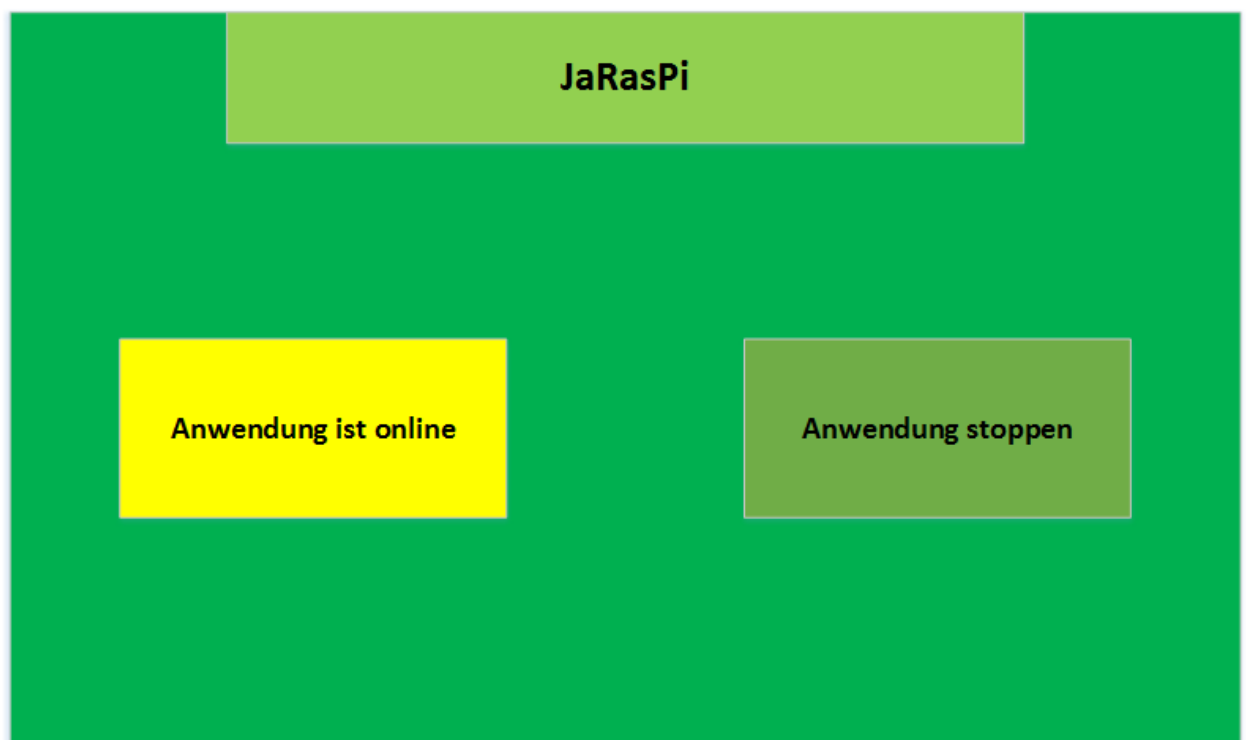


Abbildung 49: JaRasPi Oberfläche nach dem Start

4 Praktischer Bezugsrahmen

4.1 Realisierung

Nachdem im Kapitel 3 die Anforderungsdefinition und der Systementwurf erläutert wurden und das System entwickelt ist, wird im Kapitel 4 die Realisierung beschrieben. Hierbei werden die wichtigsten Programmcodes dargestellt und ihre Umsetzung erläutert.

4.1.1 Java-Anwendung

4.1.1.1 Barcode-Scanner

Die Verwendung des Barcode-Scanners in einer aktiven Java-Anwendung mit einer GUI stellte sich am Anfang als schwierig heraus. Das Ziel war es den Barcode-Scanner im Hintergrund laufen zu lassen. Beim Einscannen eines Barcodes sollte die Anwendung mit Hilfe des ActionListener den Barcode aufnehmen und verarbeiten. Da sich die herkömmlichen Barcodescanner nur als Tastatur melden, konnte das Ziel nicht erreicht werden.

Das Problem wurde gelöst, indem ein Textfeld in die GUI eingebaut wurde, auf das beim Programmstart der Fokus gesetzt wird. Nach jedem Barcodescann wird der Barcode in das Textfeld eingetragen, übersetzt, in Datenbank eingetragen, aus dem Textfeld gelöscht und der Fokus wieder auf das Feld zurückgesetzt.

4.1.1.2 Multithreading

Das Multithreading ist ein wichtiges Thema bei der Entwicklung der JaRasPi-Anwendung. Neben der ständigen Überwachung, ob ein Barcode eingescannt wurde, soll das System jederzeit von dem Benutzer angehalten werden können. Beim Starten der Anwendung wurde von der Hauptklasse ein Thread (siehe Abbildung 49) erstellt, jedoch war das Anhalten der Anwendung nicht möglich.

```

private void AnwendungStarten() {
    if(ausgangswert>neuerWert){
        HauptKlasse startJob = new HauptKlasse(JaRasPiJFrame);
        startJob.setWert2(2);
        Thread serverThread = new Thread(startJob);
        serverThread.setDaemon(true);
        serverThread.start();
        neuerWert = 7;
    }
}

```

Abbildung 50: Multithreading der Klasse HauptKlasse

Das Problem wurde durch eine Reihe von while-Schleifen gelöst, die überprüfen sollen, ob eine Bedingung erfüllt ist oder nicht. In der Abbildung 50 kann ein Abschnitt aus der while-Schleifen-Kette entnommen werden.

```

while(ausgangswert>neuerWert){
    /**
     * Die erste innere while-Schleife führt
     * die ScannerKlasse aus. In dieser wird
     * der Barcode gescannt und der Wert
     * oder aktualisiert.
     */
    while(ausgangswert>neuerWert){
        barcode = getBarcode();
        if(barcode.isEmpty()){
            break;
        }
    }
}

```

Abbildung 51: While-Schleifen-Kette

Sobald die Anwendung gestartet wurde, ist der Integer-Wert „ausgabeWert“ größer als „neuerWert“. Sobald der Benutzer auf „Anwendung anhalten“ drückt, verändern sich die Werte und die while-Schleife ist nicht mehr erfüllt. Dadurch wird die Anwendung angehalten.

Sobald die Anwendung erneut gestartet wird, werden die Werte so angepasst, dass die while-Schleifen-Bedingung erfüllt ist und das Programm seine Aufgaben weiter ausführen kann.

4.1.1.3 Produkt ein- oder ausscannen?

Die primäre Aufgabe der JaRasPi-Anwendung ist das Übersetzen von Produkten und das Erstellen von Einkaufslisten, bezogen auf Produkte, die in der Produktliste als „nichtVorhanden“ markiert worden sind. Das Problem besteht jedoch darin: Wann ist ein Produkt nicht mehr vorhanden und wie soll das Programm erkennen, ob ein Produkt eingescannt oder als verbraucht ausgedruckt wird?

Dieses Problem wurde mit einem Button gelöst. Wird der Button gedrückt werden in der HauptKlasse die Werte „ausgangswert2“ und „neuerWert2“ so verändert, dass der Code in Abbildung 51 ausgeführt werden muss.

```

while(ausgangswert2>neuerWert2){
    try {
        try {
            Thread.sleep(2000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        barcode = getBCode();

        Scanner2Klasse scannerK12 = new Scanner2Klasse(barcode, jaraspijFrame);
        scannerK12.scannerJobStarten();

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        ausgangswert2 = 1;
        jaraspijFrame.btnNewButton2.setBackground(SystemColor.blue);
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}

```

Abbildung 52: Produkt einscannen

Nach dem Betätigen des Buttons hat der Benutzer zwei Sekunden Zeit den Barcode einzuscannen. Dadurch wird das Produkt in der Produktliste um die Menge 1 erhöht. Die Zeit ist aus dem Grund eingeführt, dass zum einen der Barcode-Scanner das Produkt nicht sofort erkennt und zum anderen aber, wenn der Benutzer den Button ungewollt gedrückt hat, soll sich dieser nach zwei Sekunden von alleine abwählen.

Wird der Barcode eingescannt ohne das der Button gedrückt wird, wird die Anzahl des eingescannten Produktes in der Produktliste um die Menge 1 verringert. Es wird dann der Code in Abbildung 53 ausgeführt.

```

while(ausgangswert>neuerWert){
    barcode = getBCode();
    if(barcode.isEmpty()){
        break;
    }
    else{
        ScannerKlasse scannerK1 = new ScannerKlasse(barcode, jaraspijFrame);
        scannerK1.scannerJobStarten();

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
    break;
}

```

Abbildung 53: Produkt ausscannen

Das Produkt wird als „nichtVorhanden“ markiert, wenn die Mindestmenge 1 erreicht worden ist. Wird ein neues Produkt, welches in der Produktliste nicht vorhanden ist, eingescannt, wird das Produkt in die Produktliste als „nicht-Vorhanden“ und mit der Mindestmenge 1 versehen. Hierbei ist es gleichgültig, ob der Button gedrückt wurde oder nicht, in beiden Fällen wird das Produkt als ein neues in die Produktliste aufgenommen.

4.1.1.4 Barcode übersetzen

Das Übersetzen des Barcodes ist eine knifflige Angelegenheit. Dadurch, dass keine Datenbank mit den ganzen Produkten zur Verfügung steht, muss jedes Produkt über eine Website übersetzt werden. Um das zu erreichen, wurde die Webseite „<http://www.codecheck.info/>“ und die Java-Bibliothek Jsoup von Herrn Jonathan Hedley, siehe „<https://jsoup.org/>“, verwendet. In der Abbildung 53 wird dargestellt, wie die Übersetzung des Barcodes stattfindet.

Nachdem in die URL der Barcode hinzugefügt wurde, wird aus dem HTML-Dokument ein Document-File erstellt. Danach wird nach dem h1-Element gesucht und dessen Inhalt als das Produkt übernommen.

```
url = "http://www.codecheck.info/product.search?q="+barCode+"&OK=Suchen";
Document doc;
try {
    doc = Jsoup.connect(url).get();
    Elements body = doc.select("h1");
```

Abbildung 54: Übersetzung des Barcodes

Dadurch, dass manche Produkte nicht richtig in der Datenbank eingepflegt wurden, besteht die Möglichkeit, dass sich bei manchen Produkten die Bezeichnung im h2 befindet. Deshalb wurde in der JaRasPi-Anwendung eine Möglichkeit eingebaut, nicht nur den Namen des Produktes, sondern auch die Bezeichnung zu suchen. In der Abbildung 54 wird der Code dargestellt, der in der ScannerKlasse vorhanden ist. Soll in der zukünftigen Entwicklung die Bezeichnung mit übernommen werden, muss dieser Code-Streifen nur noch auskommentiert werden.

```
//Möchte man in der Zukunft nicht nur den Namen des Produktes, sondern auch die Kategorie in die Datenbank aufnehmen, kann der auskommentierter Quellcode dazu verwendet werden.
Elements body2 = doc.select("p");
ausgabe2 = body2.toString().replaceAll("<p class=\"product-info-label\">Produktbeschreibung</p>", "").replaceAll("<p>", "").replaceAll("</p>", "").replaceAll("&", " und ");
String aausgabe2 = ausgabe2.split("<\">")[0];
```

Abbildung 55: Übersetzung des Barcodes mit der Bezeichnung

Wurde der Barcode übersetzt muss der Inhalt der Übersetzung so angepasst werden, dass dieser in die Produktliste aufgenommen werden kann. Mit Hilfe des in der Abbildung 55 dargestellten Codes, kann der Inhalt dem entsprechend angepasst werden.

```
ausgabe = body.toString().replaceAll("<h1>", "").replaceAll("</h1>", "").replaceAll("&", " und ");
```

Abbildung 56: Korrektur des Übersetzten Barcodes

4.1.1.5 Datenbankverbindung

Nachdem das Produkt übersetzt wurde, wird eine Verbindung zu der Datenbank aufgebaut. Wie die Verbindung aufgebaut und nach bestimmten Werten in den jeweiligen Tabellen gesucht werden kann, wurde aus der Internetseite „<http://www.vogella.com/tutorials/MySQLJava/article.html>“ von Herrn Vogel, Lars entnommen. In Abbildung 56 wird die Verbindung mit der Datenbank, die Selektion der Tabelle und das Suchen nach Werten dargestellt.

```
// Mit diesem Befehl, wird der Treiber für die Datenbank geladen.
Class.forName("com.mysql.jdbc.Driver");

// Als nächstes erfolgt die Verbindung der Datenbank.
conn = DriverManager
    .getConnection("jdbc:mysql://localhost/JaRasPi?",user, password);

// Durch die Statements werden die SQL-Abfragen ermöglicht.
statement = conn.createStatement();

// Als nächsts werden die Ergebnisse der SQL-Abfrage erzeugt.
resultSet = statement
    .executeQuery("select * from JaRasPi.Produktliste");

while (resultSet.next()) {
    barcode = resultSet.getString("BarCode");
    anzahl = resultSet.getInt("Anzahl");
}
```

Abbildung 57: Datenbankverbindung

Das Prinzip wird in allen Klassen angewendet in denen eine Verbindung mit der Datenbank und das Suchen nach Werten benötigt wird.

4.1.1.6 Anzahl erhöhen/verringern

Nachdem das Produkt ohne das Betätigen des Buttons eingescannt wird, wird dieses in der Produktliste um die Menge 1 verringert. Dies wird mit dem Code in der Abbildung 57 erreicht.

```
if(anzahl >1){
    preparedStatement = conn.prepareStatement("update JaRasPi.Produktliste "
        + "set IstVorhanden=?, Anzahl= Anzahl - "+anzahl1+", Bemerkung=? where BarCode=?");
    preparedStatement.setString(1, "Vorhanden");
    preparedStatement.setInt(2, anzahl1);
    preparedStatement.setString(2, "ausgewaehlt");
    preparedStatement.setString(3, barcode);
    preparedStatement.executeUpdate();
    durchläufe ++;
    barcode = "";
}
```

Abbildung 58: Anzahl verringern

Wurde jedoch der Button betätigt und ein Produkt eingescannt, erhöht der Code in Abbildung 58 die Menge des Produktes um 1.

```

preparedStatement = conn.prepareStatement("update JaRasPi.Produktliste "
    + "set IstVorhanden=?, Anzahl= Anzahl + "+anzahl1+", Bemerkung=? where BarCode=?");
preparedStatement.setString(1, "Vorhanden");
preparedStatement.setInt(2, anzahl1);
preparedStatement.setString(2, "ausgewaehlt");
preparedStatement.setString(3, barcode);
preparedStatement.executeUpdate();
barcode = "";
durchläufe ++;

```

Abbildung 59: Anzahl erhöhen

4.1.1.7 Rezeptauswahl

Da der Benutzer unter anderem auch Rezepte auswählen kann und diese mit Zutaten befüllt sind, wird auch für die ausgewählten Rezepte die Rezeptproduktliste mit benötigten Produkten befüllt. In den nächsten Abbildungen wird dargestellt, wie die Rezeptproduktliste mit Werten befüllt wird. Das Erstellen der Rezeptproduktliste war etwas kompliziert zu programmieren, da viele Selektionen und while-Schleifen verwendet wurden, um das richtige Ergebnis zu erhalten. Damit der Leser ein Verständnis bekommt wie die Erstellung der Rezeptproduktliste funktioniert, wird dieser Punkt ausführlich erklärt.

Nachdem die Verbindung mit der Datenbank stattgefunden hat, wird in der Tabelle Rezepte nach Rezepten selektiert.

```

resultSet = statement
    .executeQuery("select * from JaRasPi.Rezepte");

/**
 * Die erste while-Schleife ist für die Kontrolle der Rezept-Tabelle zuständig.
 * Es wird geschaut, ob ein Rezept ausgewählt wurde.
 */
while (resultSet.next()) {

    rezeptName = resultSet.getString("Rezeptname");
    auswahl = resultSet.getString("Auswahl");

    /**
     * Wurde ein Rezept ausgewählt, ist die Bedingung in der if-Schleife erfüllt,
     * und diese kann ausgeführt werden.
     */
    if(auswahl.equals("ausgewaehlt")){

```

Abbildung 60: Rezeptauswahl Nr.1

Es wird solange gesucht bis ein ausgewähltes Rezept gefunden wurde. Als nächstes wird in dem ausgewählten Rezept der Produktname, die Menge und die Einheit ausgewählt.

```
// Durch die Statements werden die SQL-Abfragen ermöglicht.
statement = conn.createStatement();

// Als nächstes werden die Ergebnisse der SQL-Abfrage erzeugt.
// Nachdem ein Rezept gefunden wurde, wird mit Hilfe des Rezeptnamens
// von der richtigen Tabelle selectiert.
resultSet = statement
    .executeQuery("select * from "+rezeptName);

/**
 * In der zweiten while-Schleife wird der ausgewählte Rezept bearbeitet.
 * Hierbei werden die Wert pro zeile aufgenommen und gespeichert.
 */
while (resultSet.next()) {

    produktName = resultSet.getString("Produktname");
    menge = resultSet.getInt("Menge");
    einheit = resultSet.getString("Einheit");
```

Abbildung 61: Rezeptauswahl Nr.2

Als nächstes wird in der Produktliste nach dem ausgewählten Produkt gesucht. Hierbei ist wichtig zu erwähnen, dass auch nach ähnlichen Produkten gesucht wird. Wenn in der Produktliste beispielsweise „Mineralwasser“ und in dem Rezept „Wasser“ steht, werden beide Produkte als das gleiche Produkt anerkannt.

Wurde ein Produkt erkannt wird in der while-Schleife der Integer Wert „durchläufe“ um die Menge 1 erhöht.

```
// Durch die Statements werden die SQL-Abfragen ermöglicht.
statement2 = conn.createStatement();

// Als nächstes werden die Ergebnisse der SQL-Abfrage erzeugt.
// Hierbei werden die Werte in der Produktliste nach ähnlicher Bezeichnung
// wie der Produktname in dem Rezept gesucht.
resultSet2 = statement2
    .executeQuery("select * from JaRasPi.Produktliste WHERE ArtName LIKE '%" + produktName + "%'");
durchläufe = 1;

/**
 * In der dritten while-Schleife wird nach Produkten gesucht, die ähnlich sind wie
 * Der Produktname in dem Rezept.
 */
while (resultSet2.next()) {

    artName = resultSet2.getString("ArtName");
    istVorhanden = resultSet2.getString("IstVorhanden");
    durchläufe ++;
```

Abbildung 62: Rezeptauswahl Nr.3

Wurde ein Produkt erkannt wird kontrolliert, ob dieses Produkt als „nichtVorhanden“ markiert ist. Diese Markierung bedeutet, dass das Produkt nicht vorhanden ist und somit in die Rezeptproduktliste aufgenommen werden soll.

Werden Produkte die bereits in der Rezeptproduktliste vorhanden sind gefunden (Siehe Abbildung 62.), wird ihre Anzahl zusammengezählt.

```
/**
 * Wurde ein Produkt gefunden und dieser hat nichtVorhanden als Notiz,
 * soll folgende if-Schleife ausgeführt werden.
 */
if(!istVorhanden.equals("nichtVorhanden")){

    preparedStatement2 = conn.prepareStatement("insert into JaRasPi.Rezeptproduktliste (Produktname, Menge, Einheit, Mehrfachauswahl, Bemerkung)
    + "values (?, ?, ?, ?, ?) ON DUPLICATE KEY UPDATE Menge = Menge + "+menge+", Mehrfachauswahl = 'ja'");
    preparedStatement2.setString(1, artName);
    preparedStatement2.setInt(2, menge);
    preparedStatement2.setString(3, einheit);
    preparedStatement2.setString(4, mehrfachAuswahl);
    preparedStatement2.setString(5, "nichtAusgewaehlt");
    preparedStatement2.executeUpdate();

}
```

Abbildung 63: Rezeptauswahl Nr.4

Wurde kein Produkt gefunden, ist der Integer Wert „durchläufe“ 1 geblieben. Dadurch kann die if-Schleife ausgeführt werden. Das Produkt wird in die Rezeptliste eingetragen und falls es bereits vorhanden ist, wird die Menge zusammengezählt.

```
/**
 * Wurde kein Produkt gefunden , soll folgende if-Schleife ausgeführt werden.
 */
if(durchläufe == 1){

    preparedStatement2 = conn.prepareStatement("insert into JaRasPi.Rezeptproduktliste (Produktname, Menge, Einheit, Mehrfachauswahl, Bemerkung) "
    + "values (?, ?, ?, ?, ?) ON DUPLICATE KEY UPDATE Menge = Menge + "+menge+", Mehrfachauswahl = 'ja'");
    preparedStatement2.setString(1, produktName);
    preparedStatement2.setInt(2, menge);
    preparedStatement2.setString(3, einheit);
    preparedStatement2.setString(4, mehrfachAuswahl);
    preparedStatement2.setString(5, "nichtAusgewaehlt");
    preparedStatement2.executeUpdate();

}
```

Abbildung 64: Rezeptauswahl Nr.5

Nachdem das Rezept abgearbeitet wurde, muss die Bemerkung „ausgewaehlt“ auf „nichtAusgewaehlt“ geändert werden.

```
/**
 * Nachdem alle while-Schleifen durchlaufen sind, soll das zuvor ausgewählte Rezept, auf "nichtAusgewaehlt"
 * gesetzt werden.
 */
preparedStatement = conn.prepareStatement("update JaRasPi.Rezepte set Auswahl=? where Rezeptname=?");
preparedStatement.setString(1, "nichtAusgewaehlt");
preparedStatement.setString(2, rezeptName);
preparedStatement.executeUpdate();
```

Abbildung 65: Rezeptauswahl Nr.6

4.1.1.8 Einkaufsliste erstellen

Das Erstellen der Einkaufsliste wird in drei Bereiche unterteilt. Der erste Bereich wählt die Produkte aus der Produktliste aus und fügt alle Artikel mit der Bemerkung „nichtAusgewaehlt“ in die Einkaufsliste hinzu. Wurde ein Produkt hinzugefügt, wird die Bemerkung von „nichtAusgewaehlt“ auf „ausgewaehlt“ geändert. Siehe Abbildung 65.

```
if(bemerkung.equals("nichtAusgewaehlt")){
    preparedStatement = conn.prepareStatement("insert ignore into JaRasPi.Einkaufsliste (Produktname, Menge, Einheit) "
        + "values (?,?,?)");
    preparedStatement.setString(1, produktName);
    preparedStatement.setInt(2, anzahl);
    preparedStatement.setString(3, "");
    preparedStatement.executeUpdate();

    preparedStatement1_1 = conn.prepareStatement("update JaRasPi.Produktliste "
        + "set Bemerkung=? where ArtName=?");
    preparedStatement1_1.setString(1, "ausgewaehlt");
    preparedStatement1_1.setString(2, produktName);
    preparedStatement1_1.executeUpdate();
}
```

Abbildung 66: Einkaufsliste Nr.1

Als nächstes werden die Produkte aus der Rezeptproduktliste in die Einkaufsliste hinzugefügt. Ähnlich wie bei der Produktliste werden auch hier nur die Produkte in die Einkaufsliste hinzugefügt, die mit der Bemerkung „nicht-Ausgewaehlt“ markiert sind. Siehe Abbildung 66.

Sollen Produkte in der Rezeptproduktliste vorhanden sein, die in der Einkaufsliste bereits enthalten sind, wird auch hier nur die Menge zusammengerechnet. Nachdem die Produkte in die Liste übernommen wurden, werden sie aus der Rezeptproduktliste entfernt.

```
if(bemerkung2.equals("nichtAusgewaehlt")){
    preparedStatement2 = conn.prepareStatement("insert into JaRasPi.Einkaufsliste (Produktname, Menge, Einheit) "
        + "values (?,?,?) ON DUPLICATE KEY UPDATE Menge = Menge + "+menge2+", Einheit = '"+einheit2+"'");
    preparedStatement2.setString(1, produktName2);
    preparedStatement2.setInt(2, menge2);
    preparedStatement2.setString(3, einheit2);
    preparedStatement2.executeUpdate();

    preparedStatement2_1 = conn.prepareStatement("delete from JaRasPi.Rezeptproduktliste where Produktname=? and Menge=? and Einheit=?");
    preparedStatement2_1.setString(1, produktName2);
    preparedStatement2_1.setInt(2, menge2);
    preparedStatement2_1.setString(3, einheit2);
    preparedStatement2_1.executeUpdate();
}
```

Abbildung 67: Einkaufsliste Nr.2

Als letztes werden die Produkte aus der Getränke-Tabelle in die Einkaufsliste übernommen. Hat der Benutzer Getränke auf der Webseite ausgewählt, werden sie in der Getränke-Tabelle mit „ausgewählt“ markiert.

Ähnlich wie bei der Produktliste wird auch hier nach der Bemerkung „nichtAusgewählt“ gesucht. Werden Produkte gefunden, werden sie in die Einkaufsliste übernommen. Das ausgewählte Produkt wird mit der Bemerkung „ausgewählt“ markiert.

```
if(auswahl.equals("nichtAusgewählt")){
    preparedStatement3 = conn.prepareStatement("insert ignore into JaRasPi.Einkaufsliste (Produktname, Menge, Einheit) "
        + "values (?, ?, ?)");
    preparedStatement3.setString(1, getränkName);
    preparedStatement3.setString(2, "");
    preparedStatement3.setString(3, "");
    preparedStatement3.executeUpdate();

    preparedStatement3_1 = conn.prepareStatement("update JaRasPi.Getraenke "
        + "set Auswahl=? where Getraenke=?");
    preparedStatement3_1.setString(1, "ausgewählt");
    preparedStatement3_1.setString(2, getränkName);
    preparedStatement3_1.executeUpdate();
}
```

Abbildung 68: Einkaufsliste Nr.3

4.1.1.9 Einkaufsliste aktualisieren

Das Aktualisieren der Einkaufsliste erfolgt über zwei while-Schleifen. In der ersten wird in der Einkaufsliste nach den Produkten selektiert, siehe Abbildung 68.

```
// Durch die Statements werden die SQL-Abfragen ermöglicht.
statement = conn.createStatement();

// Als nächstes werden die Ergebnisse der SQL-Abfrage erzeugt.
// Dabei werden die Werte aus derEinkaufsliste ausgewählt.
resultSet = statement
    .executeQuery("select * from JaRasPi.Einkaufsliste");

while (resultSet.next()) {
    // Es wird nach dem Produktnamen gesucht.
    produktName = resultSet.getString("Produktname");
```

Abbildung 69: Einkaufsliste aktualisieren Nr.1

In der zweiten while-Schleife wird in der Produktliste nach gleichem oder ähnlichem Produkt gesucht. Wurde ein Produkt gefunden, welches in der Spalte IstVorhanden mit „Vorhanden“ markiert ist, wird dieses aus der Einkaufsliste entfernt, siehe Abbildung 69.

```
// Durch die Statements werden die SQL-Abfragen ermöglicht.
statement2 = conn.createStatement();

// Als nächstes werden die Ergebnisse der SQL-Abfrage erzeugt.
// Nachdem der Produktname ausgewählt wurde, wird in der Produktliste nach diesem gesucht.
resultSet2 = statement2
    .executeQuery("select * from JaRasPi.Produktliste WHERE ArtName LIKE '%" + produktName + "%'");

while (resultSet2.next()) {

    istVorhanden = resultSet2.getString("IstVorhanden");

    /**
     * Wurde ein Produkt gefunden und dieses ist als Vorhanden markiert, wird dieses aus der Einkaufsliste gelöscht.
     */
    if(istVorhanden.equals("Vorhanden")){
        preparedStatement = conn.prepareStatement("delete from JaRasPi.Einkaufsliste where Produktname=?");
        preparedStatement.setString(1, produktName);
        preparedStatement.executeUpdate();
    }
}
```

Abbildung 70: Einkaufsliste aktualisieren Nr.2

4.1.1.10 Foto erstellen

Das Erstellen des Fotos wurde mit Hilfe von Herrn MCMurray, Robert verwirklicht. Auf seiner Webseite: „https://blogs.msdn.microsoft.com/robert_mcmurray/2015/06/12/simple-java-wrapper-class-for-raspistill-on-the-raspberry-pi-2/“ konnte aus dem Quellcode entnommen werden, wie die Raspberry Pi Kamera angesprochen und das Foto erstellt werden konnte.

Wichtig ist zuerst das Definieren der Variablen und der Ort von dem die Raspberry Pi Kamera aktiviert werden kann, siehe Abbildung 70.

```
private static final long anzahlBilder = 1;

// Definierter Bereich, wo sich die raspistill-Datei befindet.
// Mit der raspistill-Datei wird die Raspberry Pi Kamera aktiviert/ angesprochen
private final String raspistillOrt = "/opt/vc/bin/raspistill";

// bildTime ist die Zeit, wie lang die Kamera Zeit hat, ein Bild zu schießen.
private final int bildTime = 2000;

// Der Integer bildQualität definiert die Qualität des Bildes.
private final int bildQualität = 100;

private int bildBreite;

private int bildHöhe;

private String bildBezeichnung;

// dateiFormat definiert den Format für das zu erstellende Bild.
private String dateiFormat = "jpg";
```

Abbildung 71: Kamera Nr.1

Wie in Abbildung 71 dargestellt, muss als nächstes nur noch ein StringBuilder verwendet werden, um die Befehle zusammenzufassen und sie im Raspberry Pi ausführen.

```
if (bildBezeichnung.indexOf('.')!=-1) dateiFormat = bildBezeichnung.substring(bildBezeichnung.lastIndexOf('.')+1);

StringBuilder sb = new StringBuilder(raspistillOrt);

sb.append(" -n -bm");
sb.append(" -t " + bildTime);
sb.append(" -w " + bildBreite);
sb.append(" -h " + bildHöhe);
sb.append(" -q " + bildQualität);
sb.append(" -e " + dateiFormat);
sb.append(" -o /var/www/html/" + bildBezeichnung);

Runtime.getRuntime().exec(sb.toString());
Thread.sleep(bildTime);
```

Abbildung 72: Kamera Nr.2

4.1.2 Website

4.1.2.1 Einkaufsliste anzeigen

Das Anzeigen der Einkaufsliste wurde mit Hilfe der Webseite von Herrn Pratzner, Axel „<http://www.php-kurs.com/mysql-datenbank-auslesen.htm>“ erstellt. Hierbei ist wichtig die Datenbank anzusprechen und die Tabelle in PHP richtig darzustellen.

```
<?php
$servername = "localhost";
$username = "Pi";
$password = "Raspberry";
$dbname = "JaRasPi";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT * FROM Einkaufsliste";
$result = $conn->query($sql);
echo'<table align="center">';
    echo'<tr>';
        echo'<th><font size="5"><font color="#fff">Produktname</th>';
        echo'<th><font size="5"><font color="#fff">Menge</th>';
        echo'<th><font size="5"><font color="#fff">Einheit</th>';
    echo'</tr>';

    if ($result->num_rows > 0) {
        while($row = $result->fetch_assoc()) {
            echo'<tr>';
                echo'<td><font size="4">' . $row['Produktname'] . '</td>';
                echo'<td><font size="4">' . $row['Menge'] . '</td>';
                echo'<td><font size="4">' . $row['Einheit'] . '</td>';
            echo'</tr>';
        }
    } else {
        echo "Die Einkaufsliste ist leer ";
    }
echo'</table>';

$conn->close();
?>
```

Abbildung 73: Einkaufsliste anzeigen

4.1.2.2 Rezept auswählen

Das Verlinken eines Rezepts wurde mit Hilfe des in Abbildung 73 dargestellten PHP-Codes erreicht.

```
echo'<tr>';
    echo'<td><font size="4"><a href="Getraenk_ausgewaehlt.php?auswahl=' . $row['Getraenkname'] . '">' . $row['Getraenkname'] . '</td>';
echo'</tr>';
```

Abbildung 74: Rezept auswählen

4.1.2.3 Rezept erstellen

Hat der Benutzer ein Rezept erstellt, wird dem Webserver der Name des Rezeptes und die Kategorie geschickt. Weiterhin muss der Benutzer die jeweiligen Zutaten eintragen, welche an den Webserver gesendet werden. Nachdem der Benutzer auf „Rezept erstellen“ gedrückt hat, wird das Rezept erstellt und in die Rezept-Tabelle eingetragen. Die unten aufgeführte Abbildung 74 stellt den PHP-Code, der für die richtige Ausführung zuständig ist, dar.

```
$sql="INSERT INTO Rezepte2 (Rezeptname, Auswahl, Kategorie) VALUES ('$_POST[fname'],'nichtAusgewaehlt','$_POST[kategorie]')";
$result = $conn->query($sql);

$sql="CREATE TABLE $_POST[fname] (
    Produktname Varchar(255) NOT NULL,
    Menge Integer(10) NOT NULL,
    Einheit VARCHAR(40) NOT NULL,
    PRIMARY KEY (Produktname))";
$result = $conn->query($sql);

$sql = "INSERT INTO $_POST[fname] (Produktname, Menge, Einheit) VALUES ('$_POST[pname1]', '$_POST[menge1]', '$_POST[einheit1]')";
$result = $conn->query($sql);
$sql = "INSERT INTO $_POST[fname] (Produktname, Menge, Einheit) VALUES ('$_POST[pname2]', '$_POST[menge2]', '$_POST[einheit2]')";
$result = $conn->query($sql);
```

Abbildung 75: Rezept erstellen

4.1.2.4 Produkt entfernen

Möchte der Benutzer ein Produkt manuell aus der Einkaufsliste entfernen, kann er dies über die Webseite erreichen. Dazu wählt er in der Rubrik „Einkaufsliste bearbeiten“ „Produkt entfernen“ und wählt ein Produkt aus. Dadurch wird das Produkt aus der Einkaufsliste entfernt. Der entsprechende Code ist unten in der Abbildung 75 aufgeführt.

```
echo'<tr>';
    echo'<td><font size="2">' . $row['Produktname'] . '</td>';
    echo'<td><font size="2">' . $row['Menge'] . '</td>';
    echo'<td><font size="2">' . $row['Einheit'] . '</td>';
    echo'<td><font size="2"><a href="Produkt_entfernt.php?Produktname=' . $row['Produktname'] . '">Entfernen</td>';
echo'</tr>';
```

Abbildung 76: Produkt entfernen

4.1.2.5 Produkt hinzufügen

Möchte der Benutzer hingegen ein Produkt in die Einkaufsliste hinzufügen, ruft er die Rubrik Produkt hinzufügen auf, füllt die Felder mit Inhalt und drückt auf „Produkt hinzufügen“. Als nächstes wird das Produkt in die Einkaufsliste hinzugefügt. Der in der Abbildung 76 aufgeführte PHP-Code führt das Hinzufügen aus.

```
$sql = "INSERT INTO Einkaufsliste (Produktname, Menge, Einheit) VALUES ('$_POST[pname]', '$_POST[menge]', '$_POST[einheit]')";
$result = $conn->query($sql);
```

Abbildung 77: Produkt hinzufügen

4.2 Probleme

Während der Realisierung sind viele Probleme entstanden, die gelöst werden konnten. Einige Probleme konnten jedoch bis jetzt nicht gelöst werden.

4.2.1 Artikelbezeichnung

Eines dieser Probleme ist die Bezeichnung der Artikel. Momentan wird die Webseite „<http://www.codecheck.info/>“ dazu verwendet, den Barcode zu übersetzen. Es kann jedoch vorkommen, dass es Produkte gibt, die in der Datenbank von Codecheck nicht vorhanden sind. In dem Fall setzt die JaRasPi-Anwendung den Produktnamen auf leer, wodurch dieses in der Produktliste nicht eingepflegt wird. Dadurch wird gewährleistet, dass das Programm keine falschen oder leeren Einträge in die Produktliste hinzufügen kann.

Es kann jedoch auch vorkommen, dass die Bezeichnung des Produktes nicht richtig in der Datenbank von Codecheck gepflegt worden ist. Wie im Punkt 4.1.1.4 erwähnt, besteht die Möglichkeit nicht nur den Namen des Produktes, sondern auch die Bezeichnung zu übersetzen. Dadurch wird gewährleistet, dass das Produkt exakter in die Datenbank eingepflegt wird. Nachteil ist jedoch, dass in der Spalte „Produktname“, statt nur den Namen, die gesamte Bezeichnung des Produktes eingetragen wird.

4.2.2 Dateninkonsistenzen

Das nächste große Problem ist die Dateninkonsistenz. Während das Einscannen der Produkte keine Einheitenbezeichnung benötigt, ist sie bei den einzelnen Zutaten der auszuwählenden Rezepte mehr als notwendig. Genau hier liegt das Problem. Werden zwei verschiedene Rezepte ausgewählt, kann es vorkommen, dass in einem nur ein Teelöffel Salz benötigt wird, in einem anderen hingegen eine Prise oder gar ein Esslöffel. Das Zusammenzählen kann in dem Fall nicht funktionieren, denn ein Esslöffel ist nicht gleich Teelöffel und jedermann definiert unter einer Prise Salz eine andere Menge.

4.2.3 Austragen aus der Produktliste

Ein weiteres Ziel der JaRasPi-Anwendung neben dem Erstellen von Einkaufslisten und dem Verwalten von Rezepten, ist das Subtrahieren der ausgewählten Zutaten von vorhandenen Produkten in der Einkaufsliste. Das heißt, wenn von der Milch 2 Liter vorhanden sind und für das Rezept XY 500 ml Milch benötigt werden, sollte das Programm diese 500 ml von den 2 Litern abziehen. Dies ist jedoch nicht möglich, da zum einen wie im Punkt 4.2.2. erläutert, die Dateninkonsistenz ein Problem verursacht und zum anderen beim Einscannen der Produkte nur der Name in die Datenbank aufgenommen wird. Die Einheit und die Menge werden in der Produktliste nicht gepflegt, da die Angaben auf der Seite Codecheckt nicht immer vorhanden sind.

Deshalb wurde das Subtrahieren in der JaRasPi-Anwendung zwar am Anfang der Entwicklung als gewünscht angesehen, am Ende jedoch wieder verworfen.

4.2.4 Datendarstellung auf der Webseite

Während der Programmierphase wurde das System auf einem Windows Rechner mit einem Apache Server betrieben. Hierbei entstanden keinerlei Probleme der Darstellung von den Umlauten in der Einkaufsliste oder in der Rezeptbezeichnung. Auf dem Raspberry Pi jedoch wurden sie nicht richtig dargestellt, was zwar bei der Anzeige der Einkaufsliste keinerlei Probleme macht, das Entfernen der Produkte jedoch nicht möglich macht. Es wurden verschiedene Ansätze probiert, jedoch führten sie nicht zum gewünschten Erfolg.

4.3 Ausblick

4.3.1 Darstellung im Raspberry Pi anpassen

Wie im Kapitel 4.2.4 erläutert, funktioniert die Darstellung der Umlaute auf der Webseite, wenn sie auf der Raspberry Pi betrieben wird, nicht richtig. In der weiteren Entwicklung des JaRasPi-Systems wird das System soweit angepasst, dass die Darstellung der Umlaute und die richtige Funktion der Produktentfernung durch den Benutzer einwandfrei funktionieren kann.

4.3.2 Subtrahieren

Möchte man, dass die Zusammenzählung der Zutaten einwandfrei stattfindet, müssen die Einheiten so angepasst werden, dass sie miteinander zusammengezählt werden können. Man muss sich auf bestimmte Einheiten einigen und nur diese in den Rezepten verwenden. Erst wenn die Bezeichnung der Einheiten eindeutig ist, kann die Anwendung soweit angepasst werden, dass die Subtraktion aus der Produktliste stattfinden kann. Davor muss jedoch beim erstmaligen Einscannen eines Produktes definiert werden, welche Menge und Einheit das Produkt hat. Erst dann kann das Subtrahieren erfolgreich stattfinden.

4.3.3 Rezepte anpassen/ändern

Eine Idee, die erst nach der Fertigstellung der Anwendung entstanden ist, ist das Anpassen von eigenen Rezepten. Hierbei kann der Benutzer die von ihm erstellten Rezepte bei Bedarf anpassen, verbessern oder löschen. Diese Funktion kann in der weiteren Entwicklung hinzugefügt werden.

5 Fazit und Aussichten

5.1 Zusammenfassung

Das Erstellen von Einkaufslisten gehört für viele Menschen zum alltäglichen Leben dazu, doch bisher gibt es keine wirkliche Alternative, wie der Benutzer um diese Aufgabe erleichtert werden kann. Selbst Unternehmen nehmen sich jedes Jahr vor eine Alternative zu schaffen und scheitern meist daran, dass die Produkte die Menschen entweder nicht ansprechen oder die Kosten so hoch sind, dass das Produkt für die Endanwender nicht in Frage kommt. Während Unternehmen scheitern, versuchen andere mit Projekten wie der „ZenFridge“ oder „Der intelligente Kühlschrank 2.0“ dem Ziel, das Einkaufen zu optimieren ein Stück näher zu kommen. Jedoch beschränken sich die Projekte nur auf den Kühlschrank und bieten für den Benutzer keine optimale Lösung.

Das Internet der Dinge ist ein wichtiges Kapitel, das für alle Menschen in der Zukunft interessant sein wird. Heutzutage sind die Menschen miteinander über das Smartphone verbunden und der Schritt der Verbundenheit mit den Haushaltsgeräten wird nicht lange auf sich warten. Das kann zum Beispiel mit dem Einplatinencomputer Raspberry Pi schon heute erreicht werden. Durch die richtige Hardware und die passende Software lässt sich der kleine Raspberry Pi in ein System verwandeln, dass Produkte verwaltet, Einkaufslisten erstellt und Rezepte zum Auswählen bereitstellt. Erreicht wird es mit einer Java-Anwendung und einem Webserver, die auf dem Raspberry Pi gleichzeitig betrieben und ausgeführt werden.

Bevor die Java-Anwendung erstellt werden konnte, musste der technische Bezugsrahmen erstellt und abgearbeitet werden. Es hat sich als sehr hilfreich herausgestellt, den theoretischen Bezugsrahmen vor der Programmierung abzuarbeiten. Zwar musste nach der Realisierung das Kapitel erneut überarbeitet werden, wurde jedoch durch das Abarbeiten, ein wichtiges Grundverständnis über die Funktion des Systems aufgebaut.

Während das Erstellen des Systemkontextes und die Kurzbeschreibung der Akteure schnell erledigt werden konnte, musste bei der Beschreibung der Use Cases mehr Zeit eingeplant werden. Da das zu erstellende System anhand der Use Cases aufgebaut wurde, musste jeder einzelne Schritt genau durchdacht werden. Auch hier mussten nach der Realisierung einige Punkte geändert werden, wobei auch viele beibehalten werden konnten.

Aufbauend auf die Use Cases wurde das Anwendungsfalldiagramm und die Systemfunktionalität erstellt. Bei dem zweiten Punkt ist es wichtig, dass die Systemablaufmodelle mit großer Sorgfalt gemacht werden, denn sie stellen den Ablauf des Programms bei bestimmten Aktionen dar.

Nachdem die Anforderungsdefinition erstellt war, wurde der Systementwurf entwickelt. Neben dem Subsystem- und dem Verteilungsdiagramm, wurde die beispielhafte Benutzeroberfläche für die Webseiten und die JaRasPi-Anwendung erstellt. In der Programmierung ist es hilfreich gewesen, die Benutzeroberfläche zumindest beispielhaft erstellt zu haben, dadurch konnten die im Kapitel „Anforderungsdefinition“ erstellten Use Cases und die Systemablaufmodelle mit der Benutzeroberfläche verbunden werden, um ein besseres Verständnis des zukünftigen Aufbaus und Funktionalität der JaRasPi-Anwendung zu erhalten.

Das vierte Kapitel „Praktischer Bezugsrahmen“ konnte erst nach der Realisierung der JaRasPi-Anwendung und des Webserverns stattfinden. In diesem Kapitel werden die realisierungsspezifischen Probleme aufgeführt. Viele Schwierigkeiten und Probleme konnten durch eine intensive Recherche erfolgreich gelöst werden. Außerdem wurden viele neue Funktionen hinzugefügt, die am Anfang der Realisierung nicht eingeplant wurden.

Es gibt auch Probleme die nicht gelöst werden konnten. Es sind nicht nur programmspezifische Probleme die entstanden sind, sondern viel mehr allgemeine, die nicht auf Anhieb gelöst werden konnten. Darunter gehört unter anderem die Dateninkonsistenz, die durch das Laden der Produktbezeichnungen aus einer Webseite entsteht. Im letzten Punkt „Ausblick“ werden die Probleme wieder aufgegriffen und eine Lösungsmöglichkeit angesprochen, wie sie im Einzelnen gelöst werden können.

5.2 Schlussfolgerungen

Das Erstellen von Einkaufslisten ist zwar eine Kleinigkeit, aber auch das ist zeitaufwändig. Ein System, das den Menschen dabei unterstützen könnte Einkaufslisten zu erstellen und dabei einen Überblick über die vorhandenen und nicht vorhandenen Produkte verschafft, würde das Leben vieler Benutzer erleichtern. Vor allem in der heutigen Zeit, in der alles schnell gehen muss, würde ein solches System im Alltag Zeit ersparen. Diese Bachelorarbeit versucht ein solches System zu realisieren.

Das JaRasPi-System wurde erfolgreich erstellt. Es bietet dem Benutzer die Möglichkeit seine Produkte und Rezepte zu verwalten. Es erstellt Einkaufslisten und aktualisiert sie automatisch. Der Benutzer muss nur die Produkte einscannen und das System erledigt den Rest. Möchte der Verbraucher selbst Produkte in die Einkaufsliste eintragen, austragen oder seine eigenen Rezepte erstellen, so kann er auch diese Aufgaben ausführen. Denn das System stellt alle gewünschten Funktionen, wie auch die Einkaufsliste, auf dem Webserver dar. Zwar gibt es noch offene Punkte, doch der Hauptkern der Bachelorarbeit wurde erreicht.

Möchte man das System weiter verbessern, muss erst darüber diskutiert werden, wie hoch die Interaktion zwischen Benutzer und Anwendung sein soll. Das Ziel dieser Bachelorarbeit ist, die Interaktion auf das Starten/Anhalten und später auch auf das Hinzufügen eines weiteren Produktes zu beschränken. Möchte man, dass das Produkt mit der exakten Menge und Einheit in die Produktliste hinzugefügt wird, muss vom Benutzer mehr Interaktion mit der JaRasPi-Anwendung vorausgesetzt werden. Da in der heutigen Zeit alles schnell gehen muss, würde es bei jedem neuen Produkt eher nachteilig sein, erst die Menge einzutragen und schließlich noch die Einheit auszuwählen.

Die Einheit ist der nächste Punkt über den nachgedacht werden soll. Möchte man beispielsweise das System soweit entwickeln, dass bei einer Auswahl eines Rezeptes die Zutaten in der Produktliste überprüft werden sollen, so muss bei Vorhandensein der Produkte, die Menge der benötigten Zutat von der Menge des vorhandenen Produktes in der Produktliste abgezogen werden. Jetzt muss es aber so sein, dass die Einheit bei der Zutat und bei dem Produkt gleich sein muss, denn das Programm kann eine Prise Salz nicht von 50 Gramm Salz abziehen. Ein ähnliches Szenario stellt ein Esslöffel Salz und ein Teelöffel Salz dar. Es sind gleiche Produkte, weisen jedoch eine unterschiedliche Einheit auf. Der Punkt der Dateninkonsistenz muss gut durchdacht werden, sollte das System soweit entwickelt werden, dass die Zutaten von den vorhandenen Produkten abgezogen werden sollen.

Bezogen auf das Internet der Dinge wird der Raspberry Pi weiter an Bedeutung gewinnen. Der Einplatinencomputer wird mit jeder Version leistungsstärker. Außerdem wird der Raspberry Pi durch die wachsende Community stetig interessanter, sodass sich in Zukunft noch mehr Möglichkeiten ergeben werden. Zwar ist der Raspberry Pi momentan bei Hobbybastlern und Privatpersonen beliebt, aber auch das wird sich ändern. Firmen werden das Potential des Einplatinencomputers erkennen und sie noch intensiver für ihre Projekte einsetzen.

Literaturverzeichnis

- Achour, Mehdi, et al.** PHP-Handbuch. *Was ist PHP?* [Online] [Zitat vom: 19. 07 2016.] <https://secure.php.net/manual/de/intro-what-is.php>.
- Adafruit.** Amazon. *2x13 Pin Header für Raspberry Pi* . [Online] [Zitat vom: 19. 07 2016.] https://www.amazon.de/gp/product/B00DDZD7MM/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1.
- Amazon.** Amazon. *Finish Dishwasher Detergent Dash Button*. [Online] [Zitat vom: 02. 05 2016.] http://www.amazon.com/gp/product/B010RFY4AO/ref=s9_acsd_al_bw_c_odsbnof_3?pf_rd_m=ATVPDKIKX0DER&pf_rd_s=merchandise-search-3&pf_rd_r=0M626DJ0AK0ZJQC52H7T&pf_rd_t=101&pf_rd_p=2452416802&pf_rd_i=10667898011.
- Amazonfresh.** Amazonfresh. [Online] [Zitat vom: 02. 05 2016.] <https://fresh.amazon.com/dash/>.
- Andelfinger, Volker und Hänisch, Till. 2015.** *Internet der Dinge - Technik, Trends und Geschäftsmodelle*. Wiesbaden : Springer, 2015.
- Aukru .** Amazon. *Aukru 3er Set Alu Kühlkörper in verschiedenen Größen für Raspberry Pi 3 Model B Raspberry pi 2 Model B zum Aufkleben*. [Online] [Zitat vom: 20. 05 2016.] http://www.amazon.de/Aukru-K%C3%BChlk%C3%B6rper-verschiedenen-Raspberry-Aufkleben/dp/B00ILK6DMA?ie=UTF8&psc=1&redirect=true&ref_=oh_aui_detailpage_o06_s00.
- Borchert, Thomas. 2000.** Heise online. *Der PC im Kühlschrank - mehr als nur gut gekühlt*. [Online] 06. 12 2000. [Zitat vom: 26. 04 2016.] <http://www.heise.de/newsticker/meldung/Der-PC-im-Kuehlschrank-mehr-als-nur-gut-gekuehlt-32442.html>.
- Bundesverband Öffentlicher Banken Deutschlands, VÖB, e.V. 2008.** <http://www.basel2.voeb.de>. *VOB Basel II*. [Online] Bundesverband Öffentlicher Banken Deutschlands, VÖB, e.V., 12. 1 2008. [Zitat vom: 31. 12 2008.] <http://www.basel2.voeb.de/index.htm>.

Christe, Jan. 2002. Heise online. *Multimedia für den Kühlschrank*. [Online] 16. 10 2002. [Zitat vom: 26. 04 2016.] <http://www.heise.de/newsticker/meldung/Multimedia-fuer-den-Kuehlschrank-68036.html>.

Dirscherl, Hans-Christian. 2014. pcwelt. *Handscanner mit Amazon-Konto*. [Online] 07. 04 2014. [Zitat vom: 02. 05 2016.] http://www.pcwelt.de/news/Amazon_Dash___Lebensmittel_direkt_aus_dem_Kuehlschrank_bestellen-Handscanner_mit_Amazon-Konto-8652587.html.

ekitchen. ekitchen. Ice Age: Die Idee vom vernetzten Kühlschrank. [Online] [Zitat vom: 14. 06 2016.] <http://www.ekitchen.de/kuechengerate/kuehlschrank/bilder/ice-age-die-idee-vom-vernetzten-kuehlschrank-14705.html#7>.

Fuchs, Jochen G. 2015. t3n. *Jetzt kommt er endlich, der selbstauffüllende Kühlschrank: Amazon startet Dash-Service*. [Online] 01. 10 2015. [Zitat vom: 02. 05 2016.] <http://t3n.de/news/amazon-dash-service-644943/>.

Golem.de. Golem.de. Raspberry Pi. [Online] [Zitat vom: 20. 05 2016.] <http://www.golem.de/specials/raspberry-pi/>.

Grün, Frank-Olive. 2014. ekitchen. *Der vernetzte Kühlschrank: Darum lässt er auf sich warten*. [Online] 25. 11 2014. [Zitat vom: 27. 04 2016.] <http://www.ekitchen.de/kuechengerate/kuehlschrank/ratgeber/vernetzter-kuehlschrank-wlan-14499.html>.

Habelitz, Hans-Peter. 2015. *Programmieren lernen mit Java*. Bonn : Rheinwerk Verlag GmbH, 2015. 978-3-8362-3517-4.

Horchert, Judith. 2015. Spiegel. *Dash Buttons für Waren-Bestellung: Amazon verdrückt sich*. [Online] 01. 04 2015. [Zitat vom: 02. 05 2016.] <http://www.spiegel.de/netzwelt/web/dash-button-amazon-irritiert-mit-bestellknoepfen-fuer-den-haushalt-a-1026652.html>.

Jacobs, Ian. 2014. Open Web Platform Milestone Achieved with HTML5 Recommendation. *Next Generation Web Technologies Build on Stable Foundation*. [Online] W3C, 28. 10 2014. [Zitat vom: 19. 07 2016.] <https://www.w3.org/2014/10/html5-rec.html.en>.

kabelbude.eu. Amazon. HDMI-Verbindungskabel HDMI-Stecker (A) auf HDMI-Stecker (A), gold plated, Länge 2,0 m. [Online] [Zitat vom: 20. 05 2016.] http://www.amazon.de/HDMI-Verbindungskabel-HDMI-Stecker-gold-plated-L%C3%A4nge/dp/B000NI4NGQ/ref=sr_1_9?s=computers&ie=UTF8&qid=1463823990&sr=1-9&keywords=hdm+ Kabel+2m.

Kasten, Philipp. Der intelligente Kühlschrank. Der intelligente Kühlschrank 2.0. [Online] [Zitat vom: 02. 05 2016.] <http://intelligenter-kuehlschrank.jimdo.com/unser-projekt/>.

Kofler, Michael. 2014. *Linux - Das umfassende Handbuch*. Bonn : Galileo Press, 2014. 978-3-8362-2591-5.

- Kuman.** Amazon. *Kuman 3.5 inch 320*480 Resolution Touch Screen TFT LCD Display Module SPI Interface with Touch Pen for Raspberry-pi Model 2 3 B/B+ 2B SC06*. [Online] [Zitat vom: 19. 07 2016.] https://www.amazon.de/gp/product/B01CNLYL1C/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1.
- Lavars, Nick.** 2016. gizmag. *Samsung's new smart fridge lets you check in on its contents through internal cameras*. [Online] 05. 01 2016. [Zitat vom: 27. 04 2016.] <http://www.gizmag.com/samsung-family-hub-smart-fridge/41192/>.
- Lubkowitz, Mark.** 2007. *Webseiten programmieren und gestalten*. Bonn : Galileo Press, 2007. 9783898428132.
- Media Markt.** Media Markt. *SAMSUNG microSDHC EVO + Adapter MB-MP32DA-EU microSDHC 32 GB*. [Online] [Zitat vom: 20. 05 2016.] http://www.mediemarkt.de/de/product/_samsung-microsdhc-evo-adapter-mb-mp32da-eu-1848354.html.
- Mehler-Bicher, Anett und Steiger, Lothar.** 2012. *Trends in der IT*. Mainz : s.n., 2012. 978-3-00-039317-4.
- Müller, Gerrit.** 2014. Fabletics. *6 clevere Tipps für einen gesunden Kühlschrank-Inhalt*. [Online] 27. 11 2014. [Zitat vom: 02. 06 2016.] <http://blog.fabletics.de/6-clevere-tipps-fur-einen-gesunden-kuhlschrank-inhalt/>.
- Neitzke, Dirk.** 2016. Statista. *Statistiken und Studien zum Thema Smartphones*. [Online] 06. 01 2016. [Zitat vom: 17. 05 2016.] <http://de.statista.com/themen/581/smartphones/>.
- Oestereich, Bernd.** 2009. *Analyse und Design mit UML 2.3*. München : Oldenbourg, 2009. 978-3-486-58855-2.
- ohneVerfasser(a).** 2006. Wikipedia. *Linux*. [Online] 15. 01 2006. [Zitat vom: 17. 05 2016.] <https://upload.wikimedia.org/wikipedia/commons/3/35/Tux.svg>.
- ohneVerfasser(b).** 2015. Wikipedia. *File:Java logo.png*. [Online] 21. 11 2015. [Zitat vom: 18. 05 2016.] https://en.wikipedia.org/wiki/File:Java_logo.png.
- Pasolini, Antonio.** 2013. gizmag. *Samsung's T9000 fridge with 10-inch touchscreen and Evernote*. [Online] 15. 01 2013. [Zitat vom: 27. 04 2016.] <http://www.gizmag.com/samsung-t9000-refrigerator-ces2013/25729/>.
- Petri, Björn und Petri, Britta.** Java-Tutorial.org. *Eigenschaften von Java*. [Online] [Zitat vom: 14. 06 2016.] <http://www.java-tutorial.org/java-eigenschaften.html>.
- Pi.** Amazon. *Raspberry PI Modell 2B Quad Core Starter Bundle mit 16GB SD Card, WLAN Stick, HDMI + LAN*. [Online] [Zitat vom: 20. 05 2016.] http://www.amazon.de/Raspberry-Modell-Starter-Bundle-Stick/dp/B00T7OHE9A/ref=sr_1_1?s=computers&ie=UTF8&qid=1463749524&sr=1-1&keywords=Raspberry+PI+Modell+2B#productDetails.
- Pratzner, Axel.** 2016. html-seminar. *CSS3 lernen: Tutorial über die Möglichkeiten mit CSS3*. [Online] 29. 06 2016. [Zitat vom: 19. 07 2016.] <http://www.html-seminar.de/css3.htm>.

Raspberry. Amazon. *Raspberry Pi 3 Model B, Quadcore CPU, 1GB RAM, WLAN, BT.* [Online] [Zitat vom: 20. 05 2016.] http://www.amazon.de/Raspberry-Pi-Model-Quadcore-WLAN/dp/B01CCOXV34?ie=UTF8&psc=1&redirect=true&ref_=oh_aui_detailpage_o05_s00.

—. Amazon. *Kamera Modul für Raspberry Pi.* [Online] [Zitat vom: 19. 07 2016.] https://www.amazon.de/gp/product/B00E1GGE40/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1.

Samsung. Samsung. *Home has a new hub.* [Online] [Zitat vom: 27. 04 2016.] <http://www.samsung.com/us/explore/family-hub-refrigerator/>.

SB Components . Amazon. *Premium Case für Raspberry Pi 3 Modell B Quad Core und Raspberry Pi 2 Modell B + (B PLUS).* [Online] [Zitat vom: 20. 05 2016.] http://www.amazon.de/Premium-Case-Raspberry-Pi-Modell/dp/B00MQLB1N6?ie=UTF8&psc=1&redirect=true&ref_=oh_aui_detailpage_o05_s00.

sb.component. Ebay. *Neu NOIR Raspberry Pi Kamera V2 8M Klar Kameragehäuse by SB Components.* [Online] [Zitat vom: 19. 07 2016.] http://www.ebay.de/itm/252392590770?_trksid=p2060353.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT.

Sonntag, Christian. 2012. Hochschule Niederrhein. *„Die Milch läuft morgen ab“: Studierende entwickeln intelligenten Kühlschrank.* [Online] 08. 03 2012. [Zitat vom: 02. 05 2016.] <https://www.hs-niederrhein.de/news/news-detailseite/die-milch-laeuft-morgen-ab-studierende-entwickeln-intelligenten-kuehlschrank-3606/>.

TaoTronics. Amazon. *TaoTronics USB Laser-Barcodescanner .* [Online] [Zitat vom: 19. 07 2016.] https://www.amazon.de/TaoTronics-TT-BS014-USB-Laser-Barcodescanner/dp/B00FEALRWG/ref=pd_bia_yo_i_3?ie=UTF8&psc=1&refRID=H1YF0REZB8F3S40FMTXV.

Thommes, Ferdinand. 2016. Golem.de. *Linux 4.6 mit neuem Dateisystem OrangeFS.* [Online] 17. 05 2016. [Zitat vom: 17. 05 2016.] <http://www.golem.de/news/kernel-linux-4-6-mit-neuem-dateisystem-orangefs-1605-120918.html>.

Valadur® Supplies. Amazon. *Rydgges Premium High-Quality PC Netzteil 2A / 2000 mAh Ladegerät 5V 2000 mAh für Raspberry Pi Modell A und B & Banana Pi - Pi Model B+ (B Plus) und Pi 2 Model B (die neueste Version 2015) Pi Xero und Pi 3 (Modell 2016).* [Online] [Zitat vom: 20. 05 2016.] http://www.amazon.de/Rydgges-Ladeger%C3%A4t-Netzteil-Ausschaltknopf-Raspberry/dp/B01C4HD986/ref=sr_1_3?ie=UTF8&qid=1463824695&sr=8-3&keywords=Netzteil+mit+USB+Kabel.

Abkürzungsverzeichnis

bzw. beziehungsweise

etc. et cetera

OS Operating System

usw. und so weiter

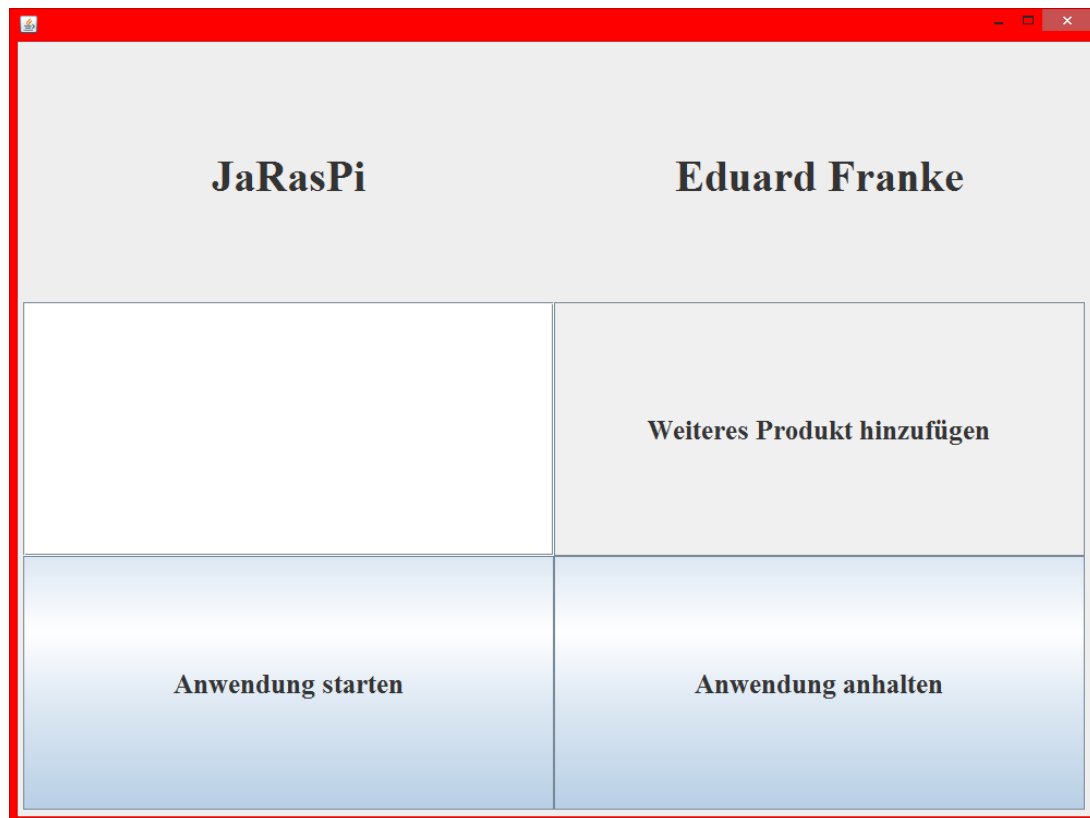
z. B. zum Beispiel

Anhang

Bedienungsanleitung

Java-Anwendung

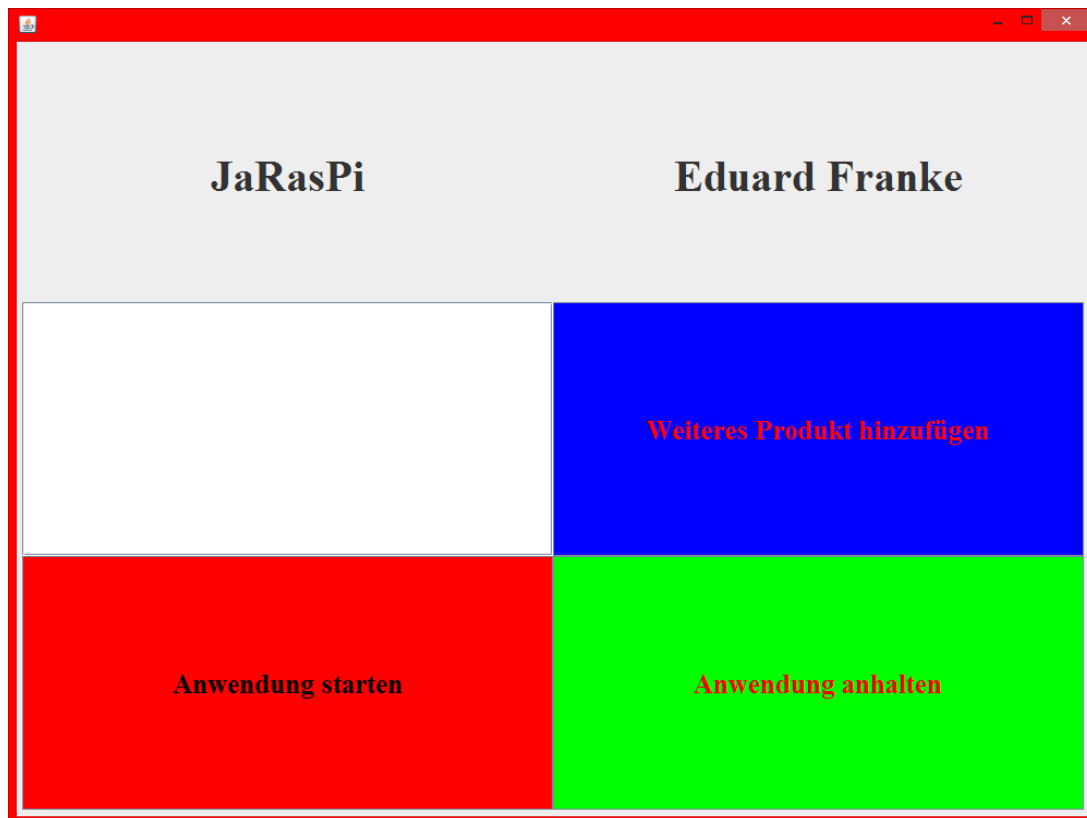
Die Verwendung der JaRasPi-Anwendung ist einfach gestaltet. Nach dem Start der Anwendung erscheint das Startfenster.



Nach dem Betätigen des „Anwendung starten“-Buttons wird die Anwendung gestartet.

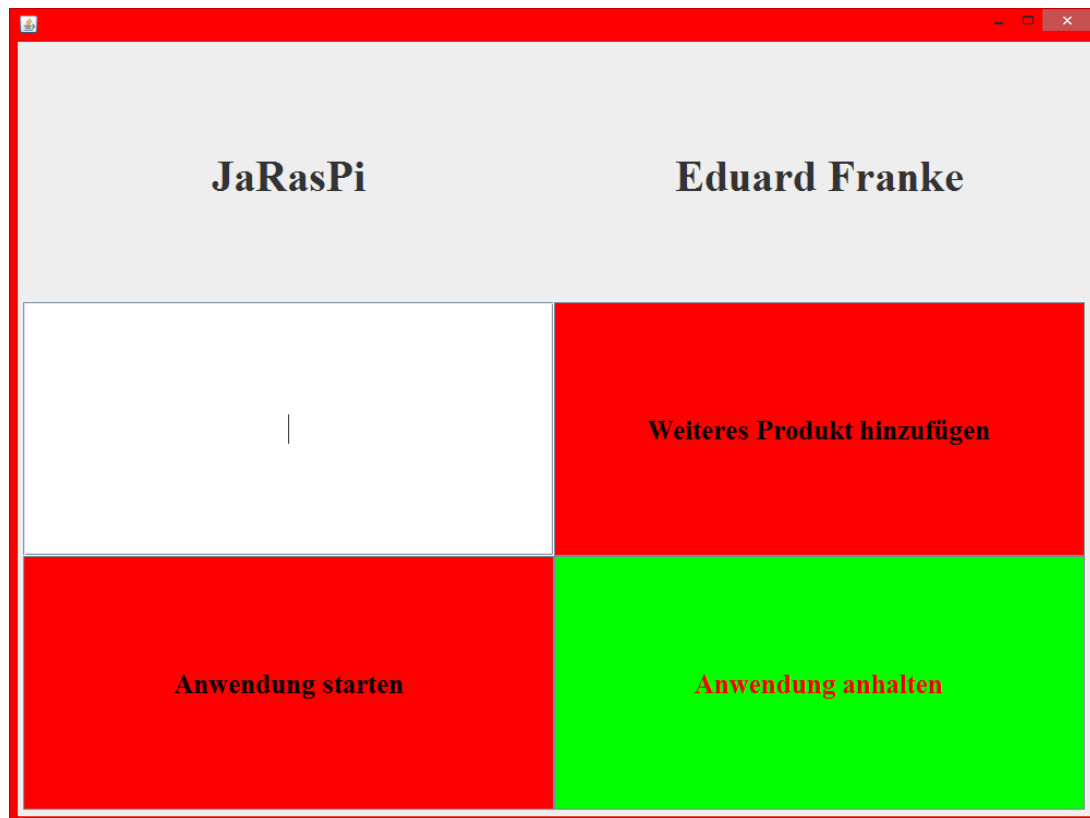
Der Barcode kann bereits eingescannt werden. Wird der Barcode jetzt eingescannt, wird die Anzahl des vorhandenen Produktes um 1 reduziert. Neue Artikel, die noch nie eingescannt wurden, werden mit einem Mindestbestand von 1 in die Datenbank aufgenommen.

Artikel, die den Mindestbestand erreichen, werden in die Einkaufsliste eingetragen.

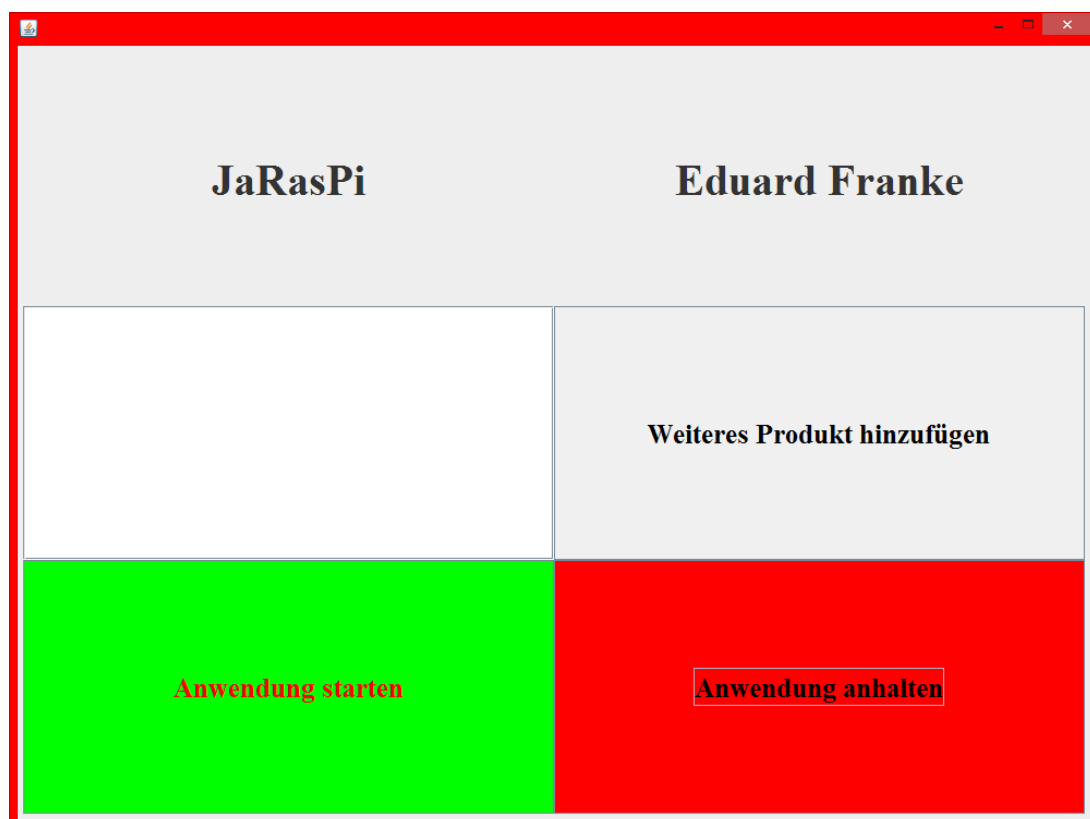


Möchte man jetzt ein Produkt, das man gekauft hat in die Datenbank als vorhanden aufnehmen, muss man den Button „Weiteres Produkt hinzufügen“ drücken. Das Ergebnis sieht wie in der unteren Abbildung dargestellt aus.

Durch das Drücken des Buttons wird dem Benutzer zwei Sekunden Zeit gegeben das Produkt einzuscannen. Währenddessen bucht das System es automatisch in die Datenbank ein. Danach wird jedes eingescannte Produkt wieder ausgebucht.

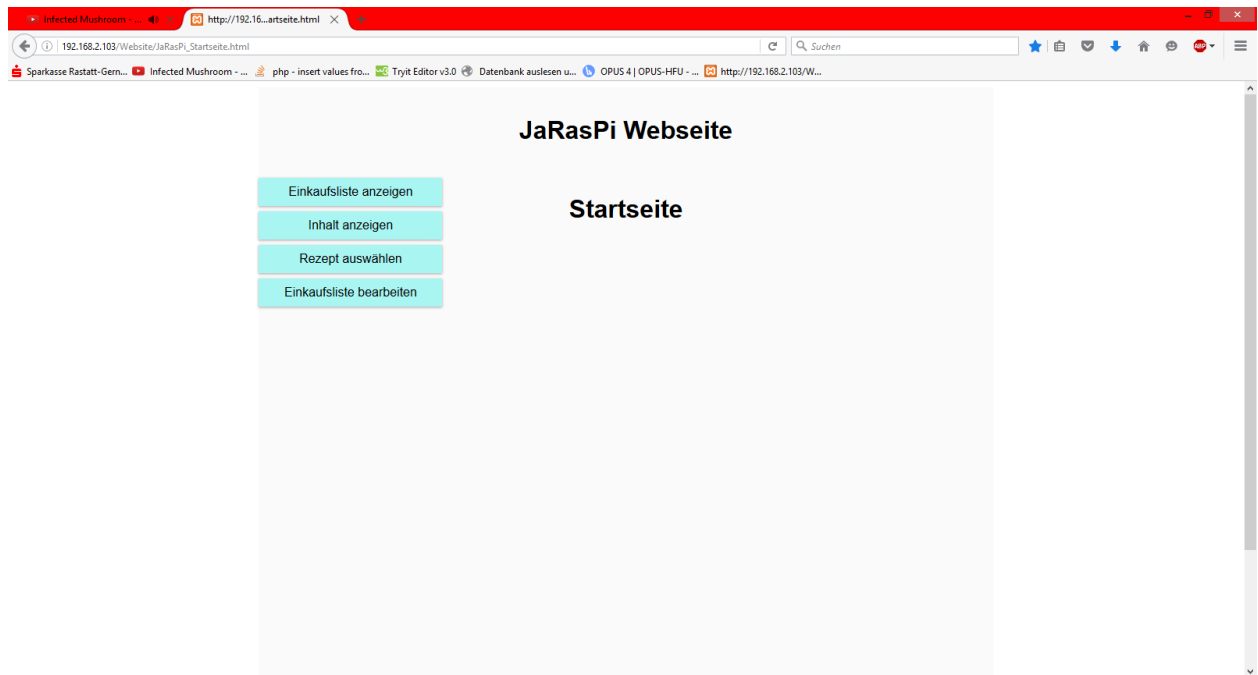


Der dritte Button ist für das Anhalten der Anwendung zuständig. Wird dieser betätigt, werden alle Aktivitäten der Anwendung angehalten.

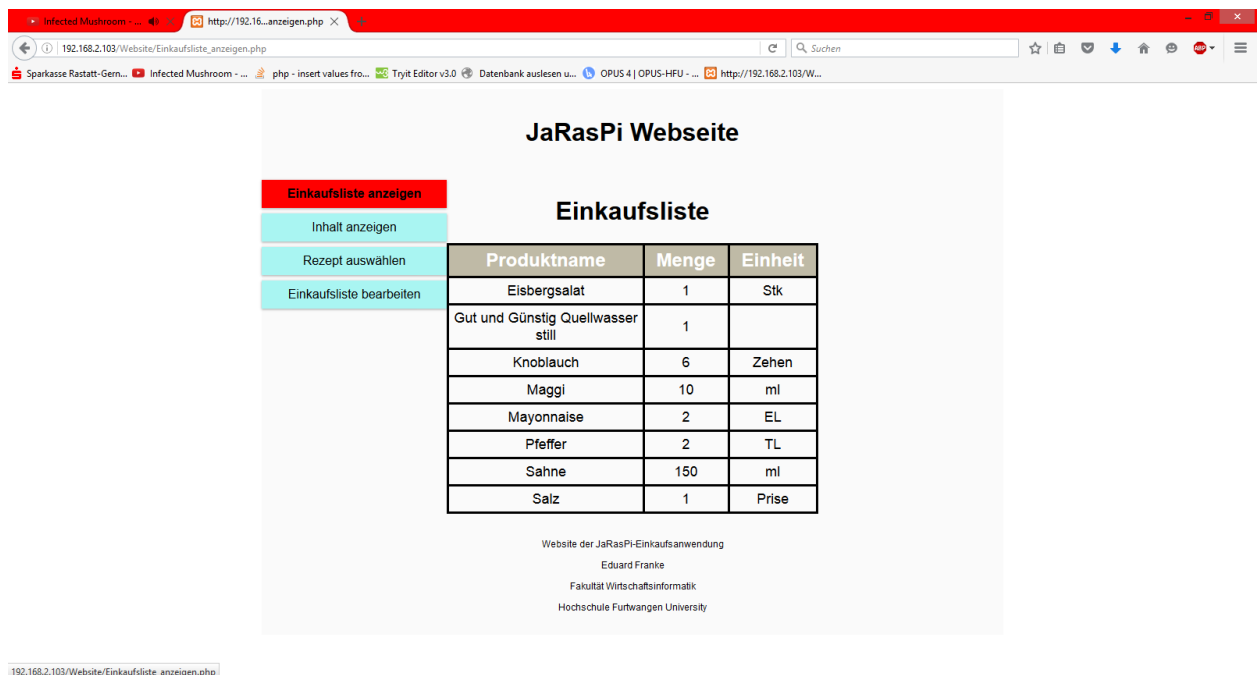


Webseite

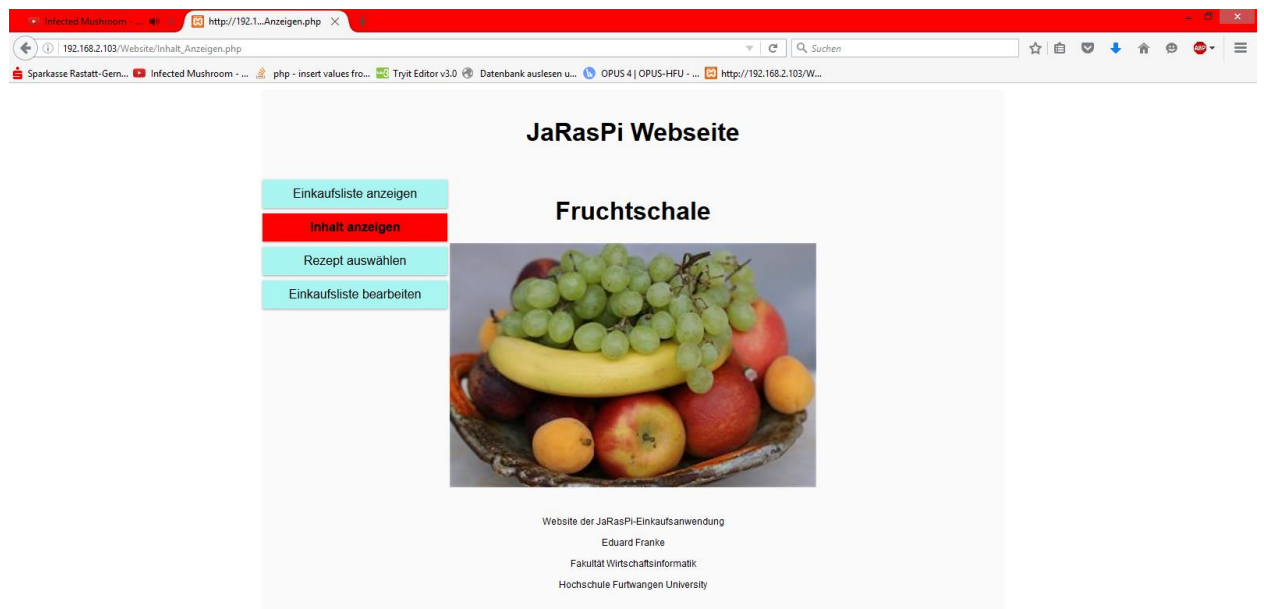
Nachdem die Webseite aufgerufen wurde, erscheint die Startseite.



Das Auswählen von „Einkaufsliste anzeigen“ stellt die digitale Einkaufsliste bereit.



Wählt der Benutzer „Inhalt anzeigen“ aus, wird das zuletzt erstellte Foto von der Fruchtschale angezeigt.



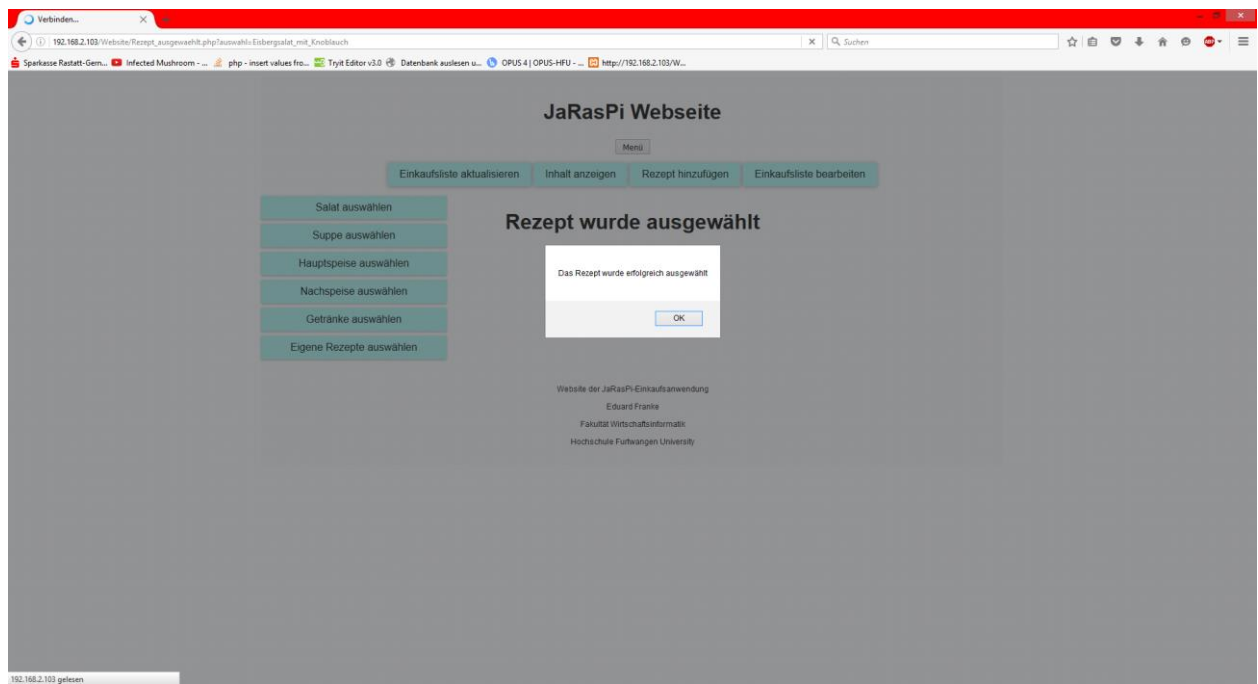
Möchte man hingegen Rezepte auswählen, so drückt man auf den „Rezept auswählen“-Button. Hier kann man sich wiederum zwischen verschiedenen Kategorien, in denen die Rezepte eingepflegt worden sind, entscheiden.



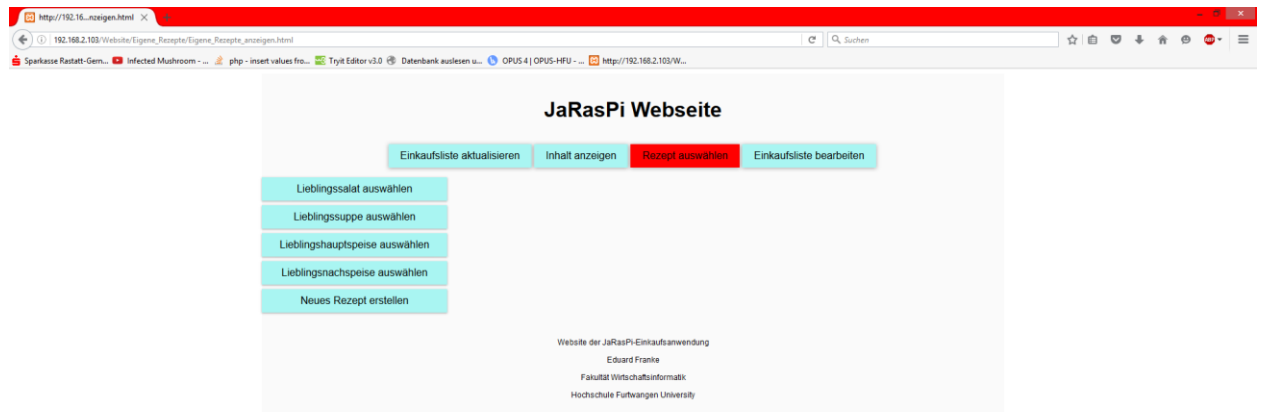
Wählt man „Salat auswählen“ aus, werden die standartmäßig vorhandenen Rezepte angezeigt.



Wurde ein Rezept ausgewählt, wird der Benutzer darüber unterrichtet, dass das Auswählen erfolgreich gewesen ist.



Wählt der Benutzer hingegen „Eigene Rezepte auswählen“ aus, gelangt man in den Abschnitt, in dem eigene Rezepte ausgewählt werden können. Eigene Rezepte, sind die Rezepte, die der Benutzer selbst anlegen kann.



Sind noch keine Rezepte eingepflegt worden, wird beim Auswählen der einzelnen Kategorien nichts angezeigt.



Möchte der Benutzer ein Rezept erstellen, so muss er den Rezeptnamen eintragen und die Kategorie auswählen. Weiterhin muss der Benutzer die Zutaten für das Rezept eintragen.

The screenshot shows a web browser window with the URL `http://192.168.2.103/entstellen.php`. The page title is "JaRasPi Webseite". At the top, there are four buttons: "Einkaufsliste aktualisieren", "Inhalt anzeigen", "Rezept hinzufügen", and "Einkaufsliste bearbeiten". On the left side, there is a vertical menu with five buttons: "Lieblingssalat auswählen", "Lieblingssuppe auswählen", "Lieblingshauptpeise auswählen", "Lieblingsnachspeise auswählen", and "Neues Rezept erstellen" (highlighted in red). The main content area is titled "Rezept erstellen". It contains two input fields: "Rezeptname" (with the value "Muffins") and "Kategorie" (with a dropdown menu showing "Nachspeise"). Below this is a section titled "Bitte Zutaten eintragen" which is a table with three columns: "Produktname", "Menge", and "Einheit". The table contains 10 rows of ingredients with their respective quantities and units.

Produktname	Menge	Einheit
Naturjoghurt	250	ml
Zucker	350	g
Öl	150	ml
Vanillezucker	20	g
Eier	3	Stk
Mehl	750	g
Backpulver	20	g
Milch	100	ml
		g
		g
		g
		g
		g

Wurden alle Daten eingetragen, kann der Benutzer das Rezept über den Button „Rezept erstellen“ in die Datenbank eintragen. Dadurch wird das Rezept erstellt und kann in der entsprechenden Kategorie ausgewählt werden.

The screenshot shows the same web browser window, but the page is now displaying a success message. The "Rezept erstellen" form is still visible, but a modal dialog box is centered on the screen with the text "Das Rezept wurde erfolgreich erstellt" and an "OK" button. Below the dialog box, the footer text is visible: "Webseite der JaRasPi-Einkaufsanwendung", "Eduard Franke", "Fakultät Wirtschaftsinformatik", and "Hochschule Fulda University".

In der unten aufgeführten Abbildung wird dargestellt, dass das Rezept erfolgreich erstellt und in die Kategorie eingetragen wurde.



192.168.2.103/Website/Eigene_Rezepte/Suesses_auswählen.php

Möchte der Benutzer die Einkaufsliste bearbeiten, so wählt er „Einkaufsliste bearbeiten“ aus. Hier kann er sich zwischen „Produkt entfernen“ oder „Produkt hinzufügen“ entscheiden.



192.168.2.103/Website/Einkaufsliste_bearbeiten.php

Wird „Produkt entfernen“ ausgewählt, kann durch das drücken auf „Entfernen“ das ausgewählte Produkt aus der Einkaufsliste entfernt werden.

JaRasPi Webseite

Aus der Einkaufsliste entfernen

Produktname	Menge	Einheit	Produkt entfernen
Eisbergsalat	1	Stk	Entfernen
Gut und Günstig Quellwasser still	1		Entfernen
Knoblauch	6	Zehen	Entfernen
Magg	10	ml	Entfernen
Mayonnaise	2	EL	Entfernen
Pfeffer	2	TL	Entfernen
Sahne	150	ml	Entfernen
Salz	1	Prise	Entfernen

Website der JaRasPi-Einkaufsanwendung
Eduard Franke
Fakultät Wirtschaftsinformatik
Hochschule Fulda University

192.168.2.103/Website/Produkt_entfernt.php?Produktname=Eisbergsalat

Nachdem drücken auf „Entfernen“ wird das jeweilige Produkt entfernt. Nach dem Ausführen des Befehls kann der Benutzer weitere Produkte entfernen.

JaRasPi Webseite

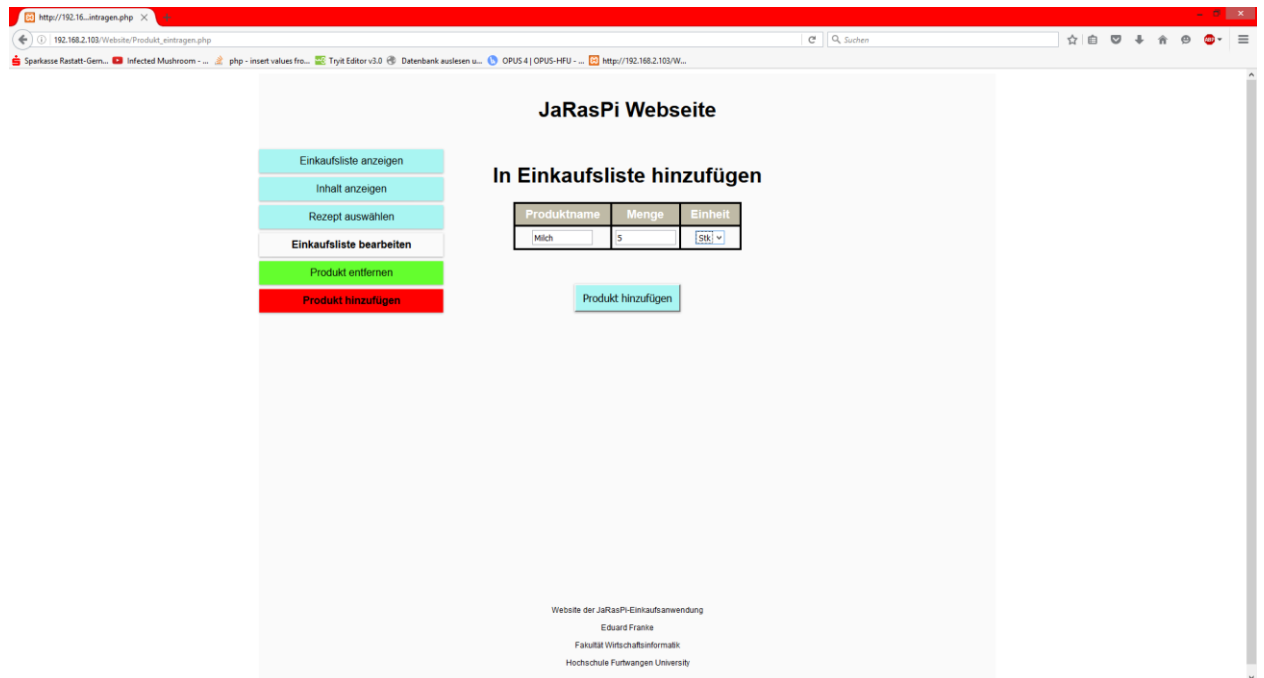
Aus der Einkaufsliste entfernen

Produktname	Menge	Einheit	Produkt entfernen
Mayonnaise	2	EL	Entfernen
Pfeffer	2	TL	Entfernen
Sahne	150	ml	Entfernen
Salz	1	Prise	Entfernen

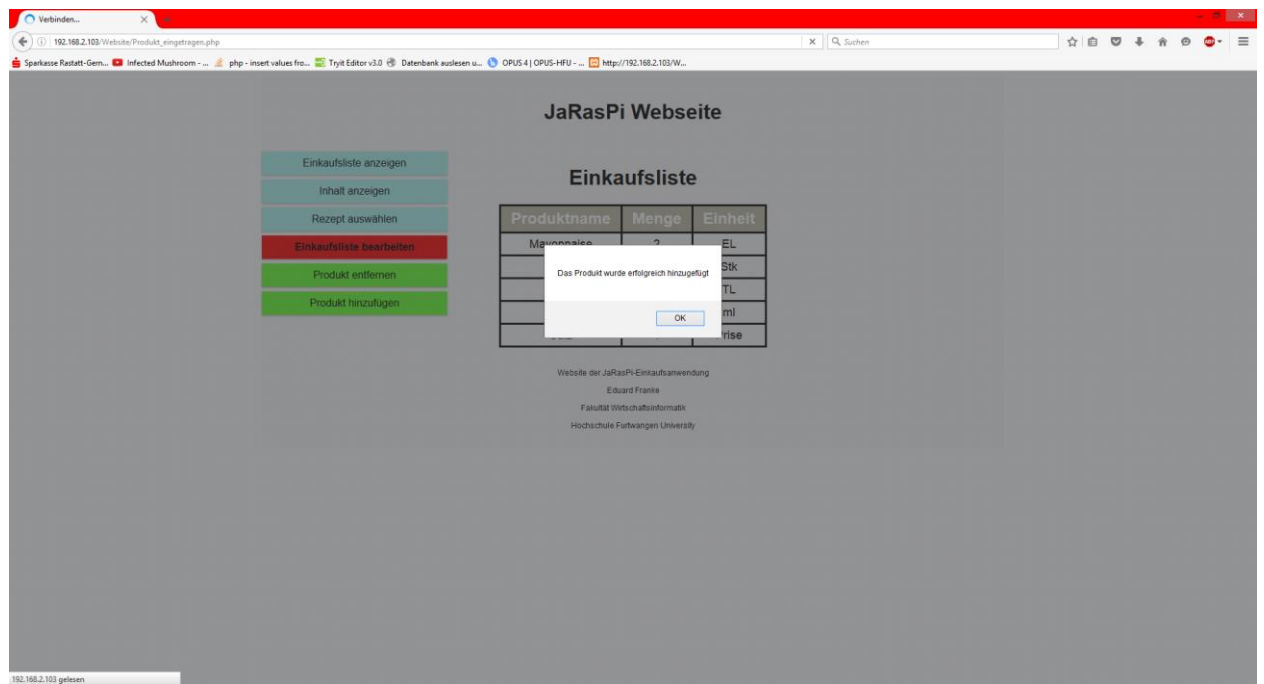
Website der JaRasPi-Einkaufsanwendung
Eduard Franke
Fakultät Wirtschaftsinformatik
Hochschule Fulda University

192.168.2.103/Website/Produkt_entfernt.php?Produktname=Mayonnaise

Wurde „Produkt hinzufügen“ ausgewählt, kann durch das Ausfüllen der Felder und das abschließende Bestätigen des Buttons „Produkt hinzufügen“ das Produkt in die Einkaufsliste hinzugefügt werden.



Nachdem Bestätigen wird der Benutzer informiert, dass das Hinzufügen erfolgreich gewesen ist.



Nach dem Schließen des Informationsfensters wird die bereits aktualisierte Einkaufsliste dem Benutzer zur Verfügung gestellt.

JaRasPi Webseite

Einkaufsliste

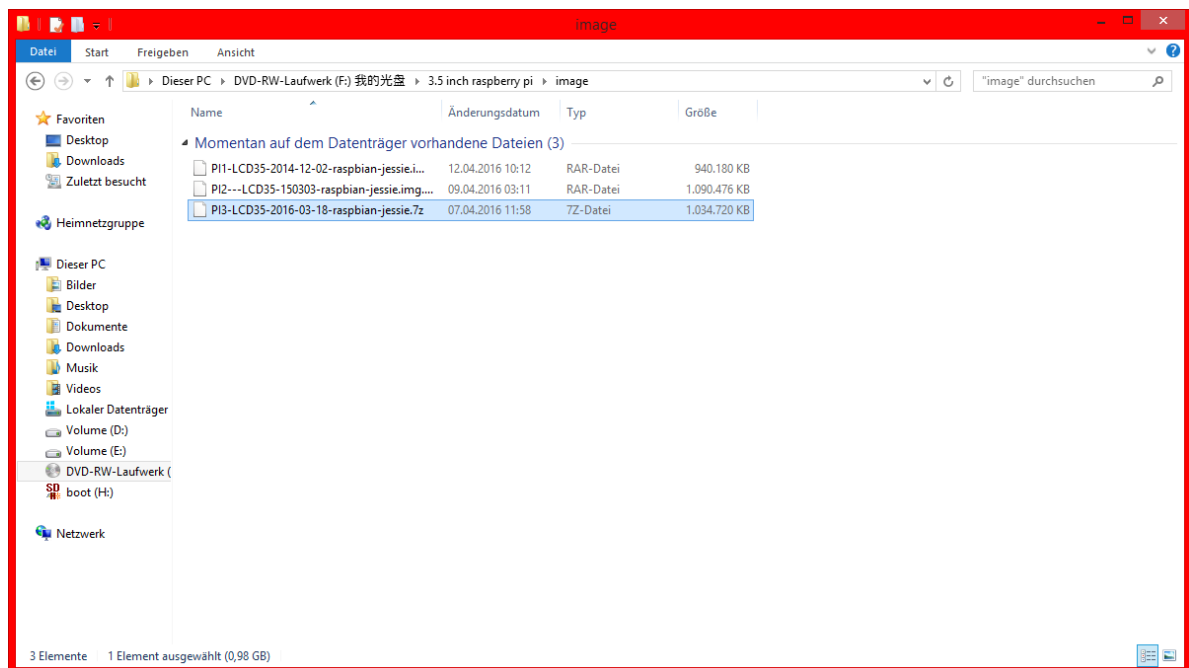
Produktname	Menge	Einheit
Mayonnaise	2	EL
Milch	5	Stk
Pfeffer	2	TL
Sahne	150	ml
Salz	1	Prise

Website der JaRasPi-Einkaufsanwendung
Eduard Franke
Fakultät Wirtschaftsinformatik
Hochschule Furtwangen University

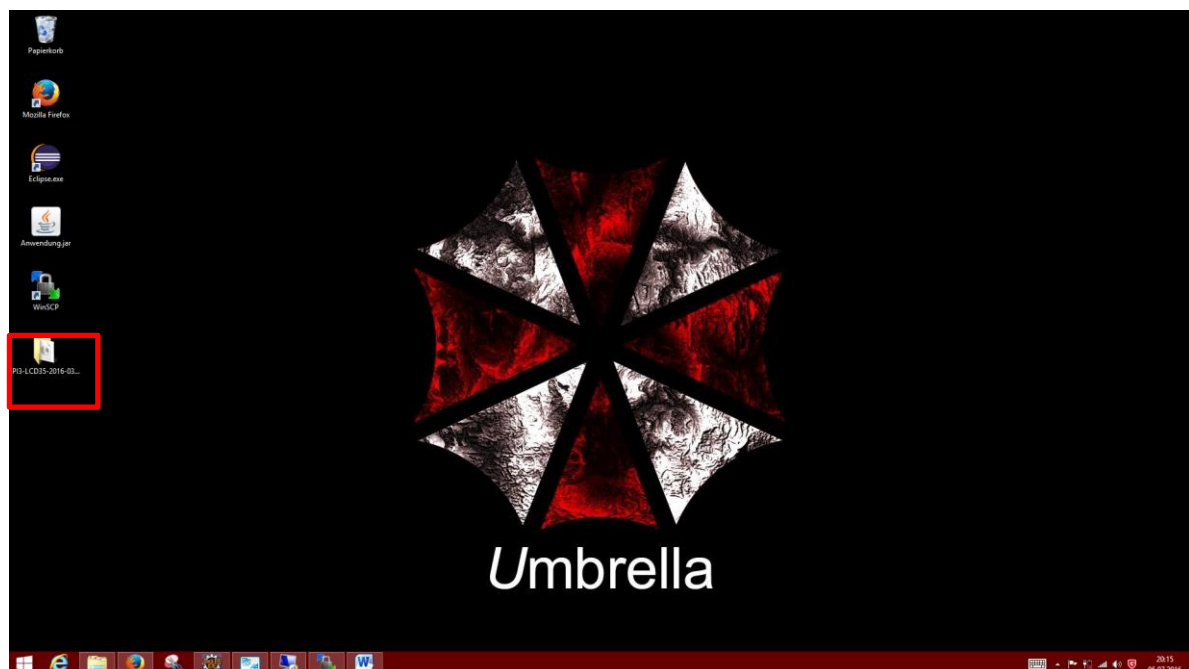
Bereitstellung der Arbeitsumgebung

Betriebssystem installieren

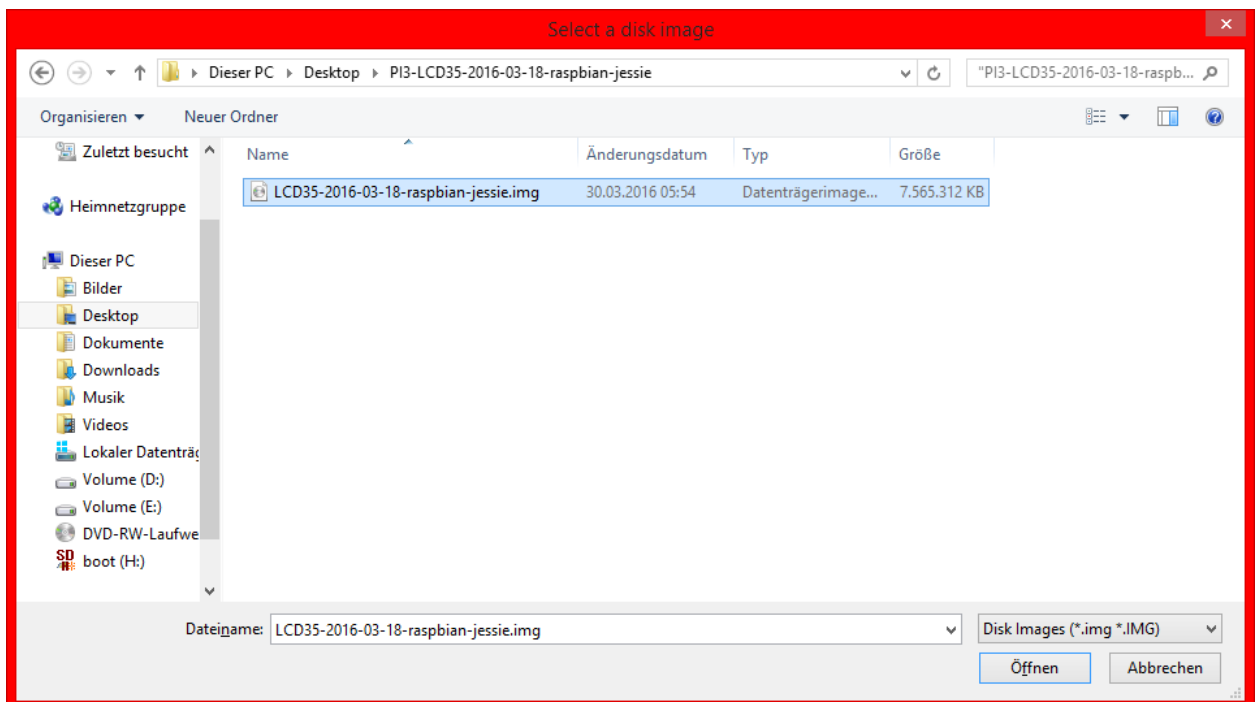
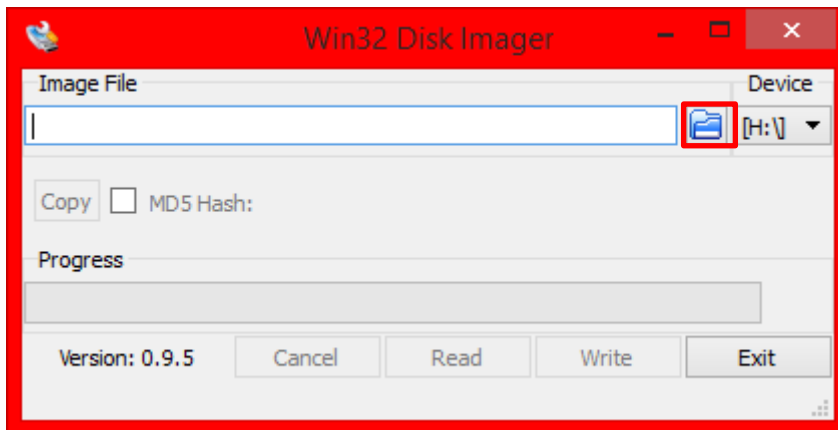
Da das System über einen 3,5 Zoll LC-Display bedient werden soll, muss das von dem Hersteller mitgelieferte Betriebssystem installiert werden. Das OS befindet sich auf einer DVD und muss erst entpackt werden.



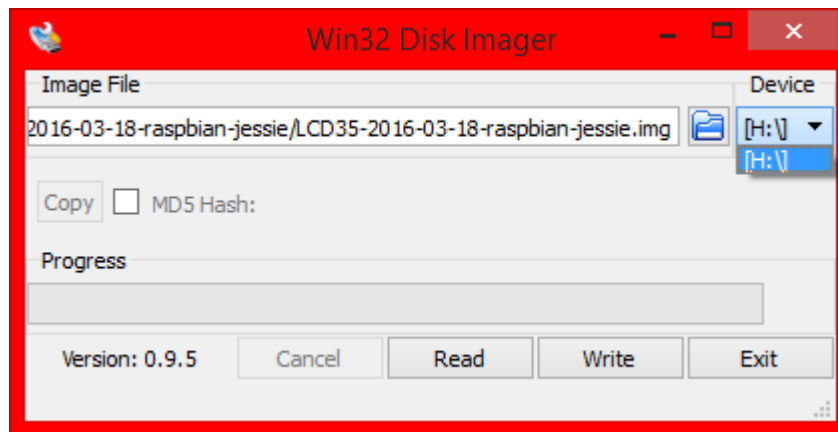
Nach dem Entpacken steht das Image-File zur Verfügung. Siehe untere Abbildung.



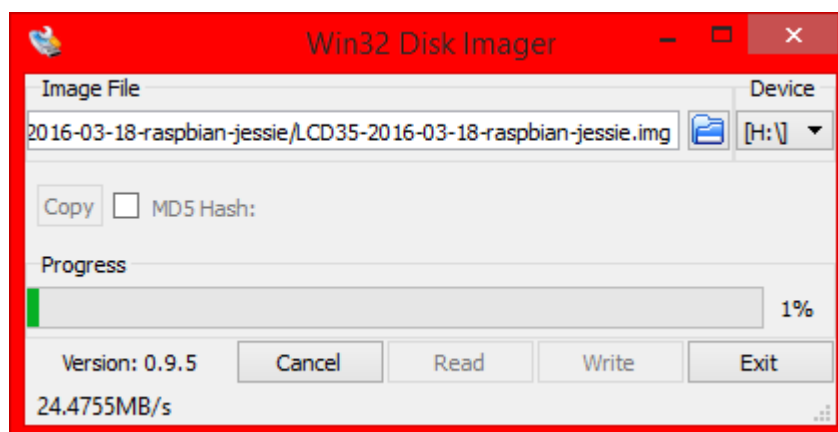
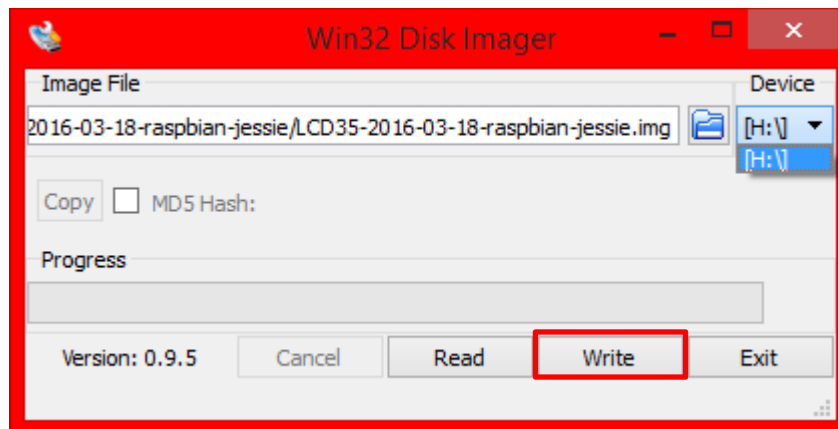
Als nächstes wird mit dem Programm „Win32 Disk Imager“ oder einem ähnlichen Programm das Image-File auf die SD-Karte geschrieben. Dazu wählt man zuerst das Image aus.



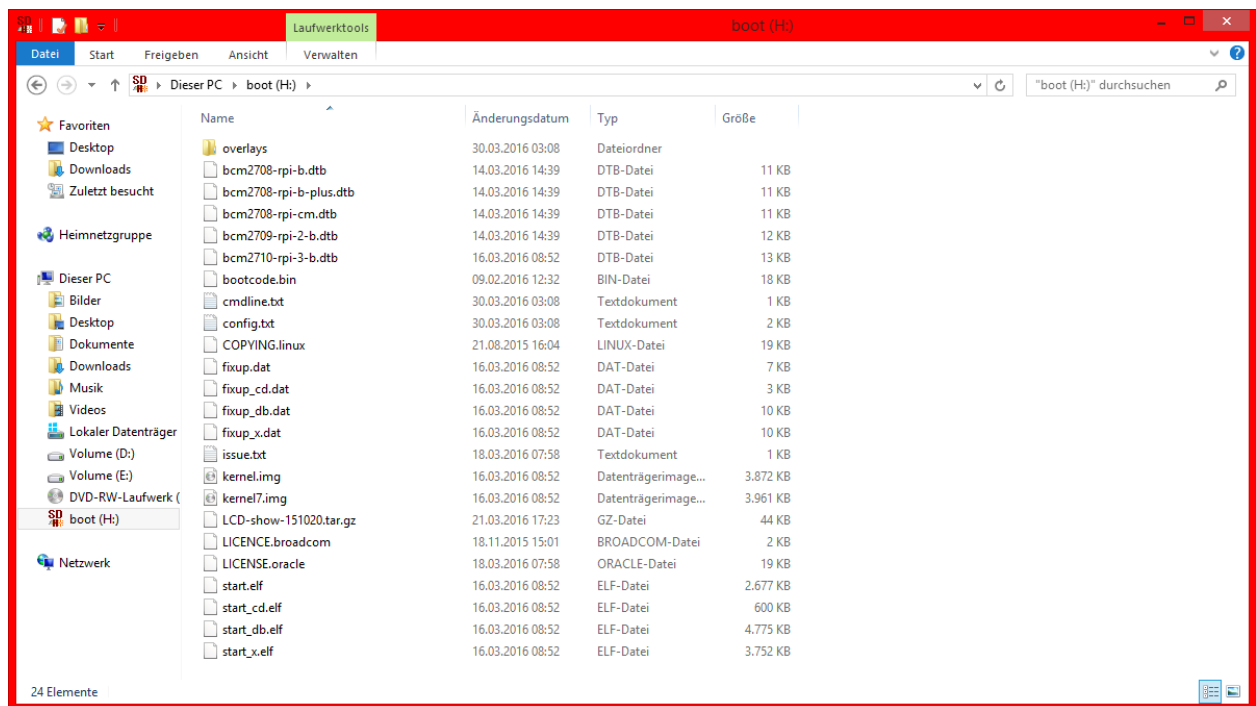
Als nächstes wird eine SD-Karte ausgewählt. Hierbei ist es wichtig, dass die richtige SD-Karte ausgewählt wird, da man sonst Gefahr läuft, die gesamten Daten auf dem ausgewählten Gerät zu verlieren.



Als letztes wird der Button „Write“ ausgewählt. Dadurch wird der Schreibvorgang aktiviert.



Nach Abschluss sieht das Ergebnis wie in der unten aufgeführten Abbildung dargestellt aus.



Nun kann die SD-Karte in den Raspberry Pi eingesetzt werden. Der Raspberry Pi ist somit einsatzbereit.

Von LCD auf HDMI

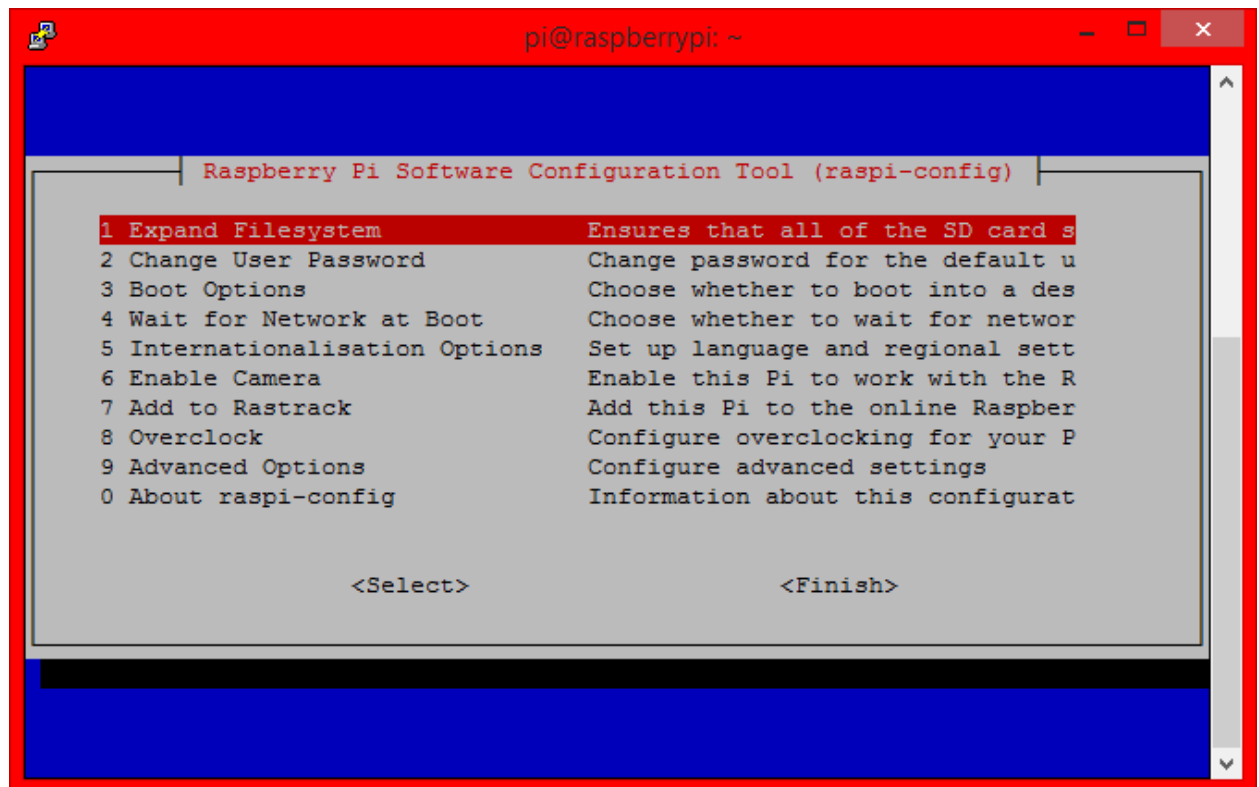
Der kleine LCD erschwert das Arbeiten auf dem Raspberry Pi. Deshalb muss auf den HDMI-Ausgang umgeschaltet werden. Dazu wird das Terminal aufgerufen und die Befehle eingetragen. Diese Befehle sind:

- `sudo cd LCD-show/`
- `sudo ./LCD-hdmi`

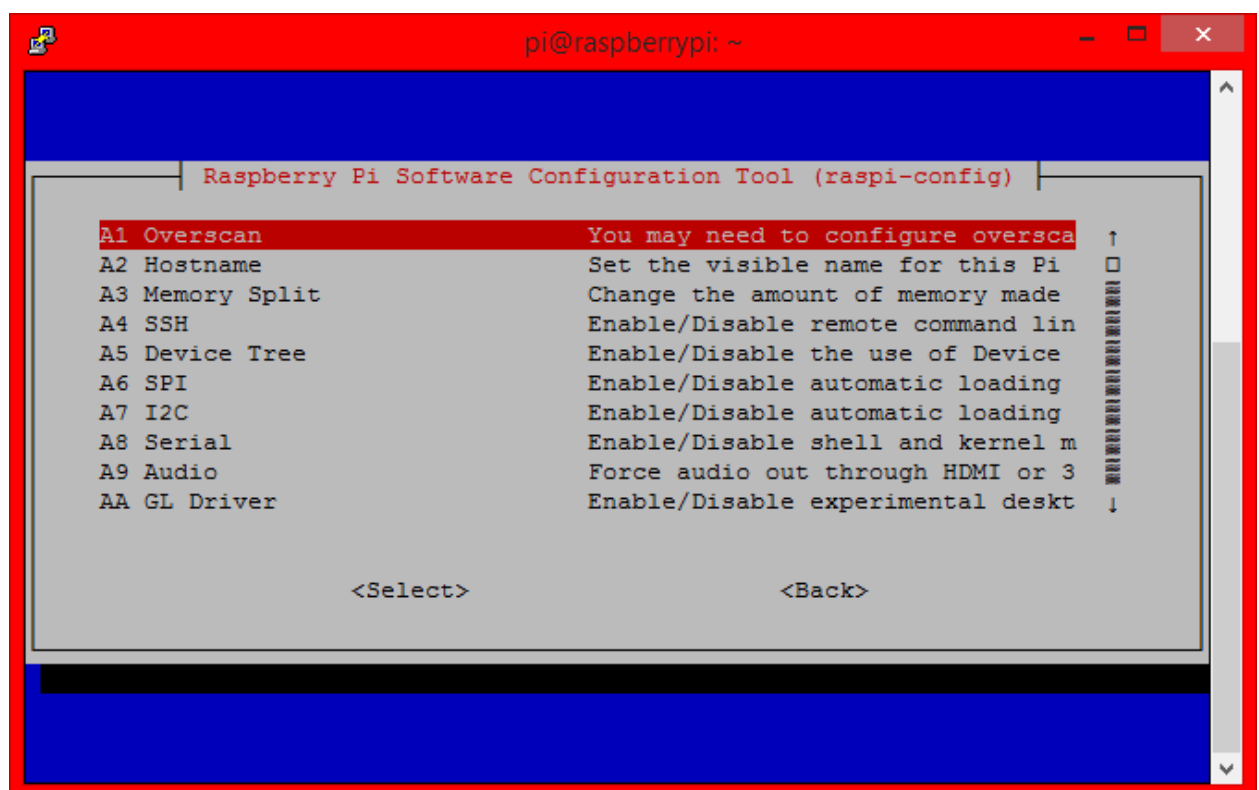
Nach dem Ausführen wird der Raspberry Pi neugestartet und der HDMI-Ausgang funktioniert.

SSH-Zugriff einrichten

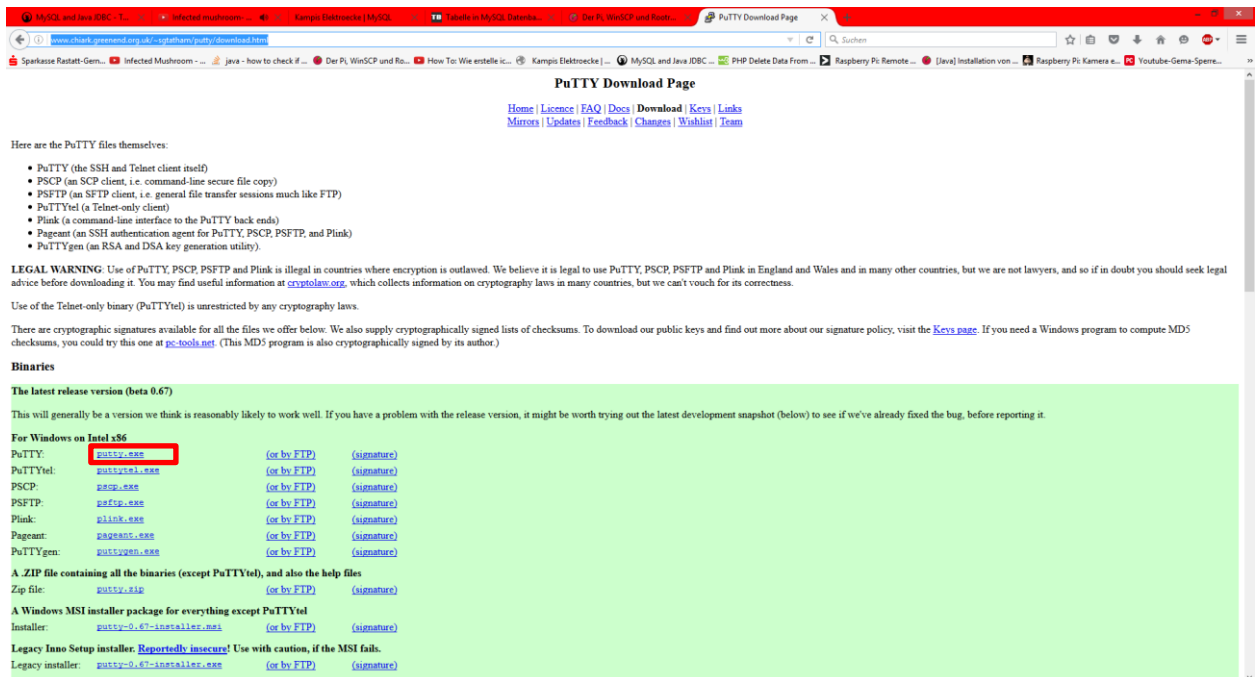
Um auf den Raspberry Pi von eigenem Computer zugreifen zu können, müssen einige Punkte abgearbeitet werden. Erstens muss auf dem Pi die SSH-Verbindung aktiviert werden. Dazu gibt man in dem Terminal den Befehl „`sudo raspi-config`“ ein.



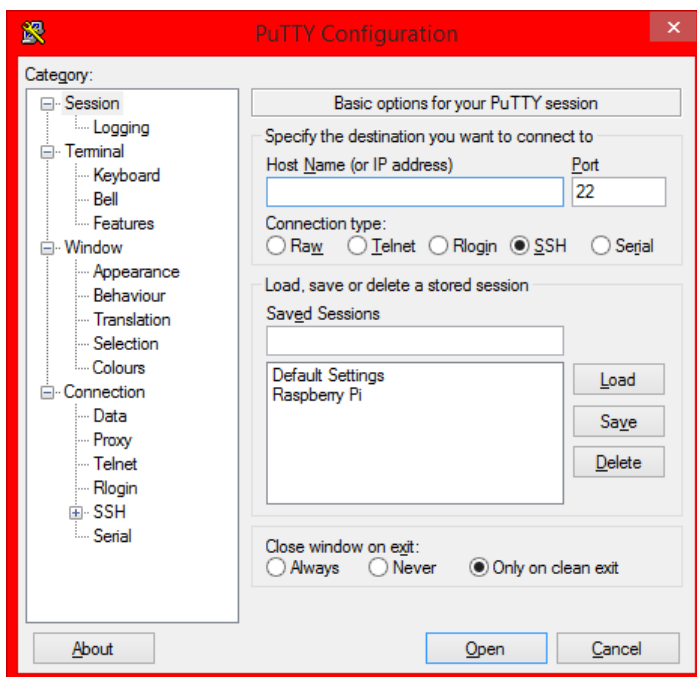
Als nächstes wählt man „Advanced Options“ aus und setzt „A4 SSH“ auf Enable.



Nachdem SSH auf Enable gesetzt wurde, kann die Verbindung zu dem Raspberry Pi mit Hilfe von Putty hergestellt werden. Putty kann auf der Website „<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>“ heruntergeladen werden. Nach dem Aufruf der Website wählt man wie in der unteren Abbildung putty.exe aus. Diese ist ohne eine Installation ausführbar.

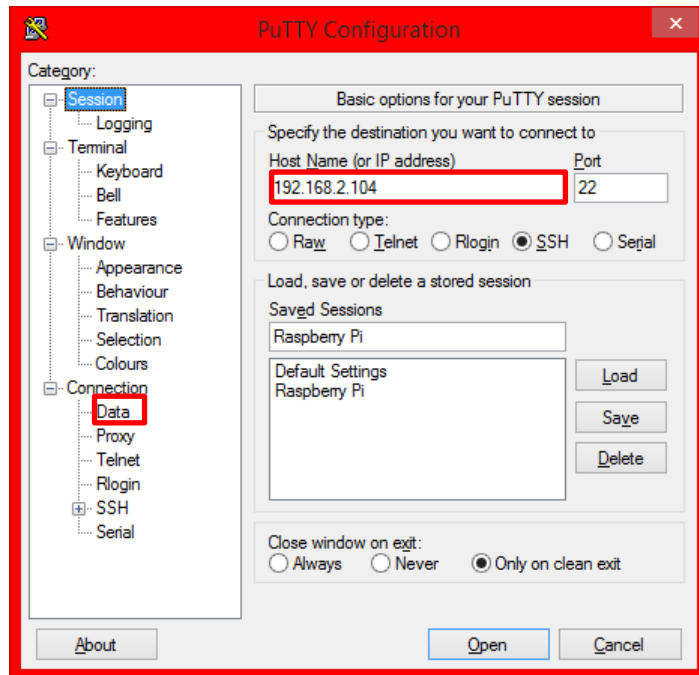


Nach dem Ausführen von Putty erscheint ein Fenster.

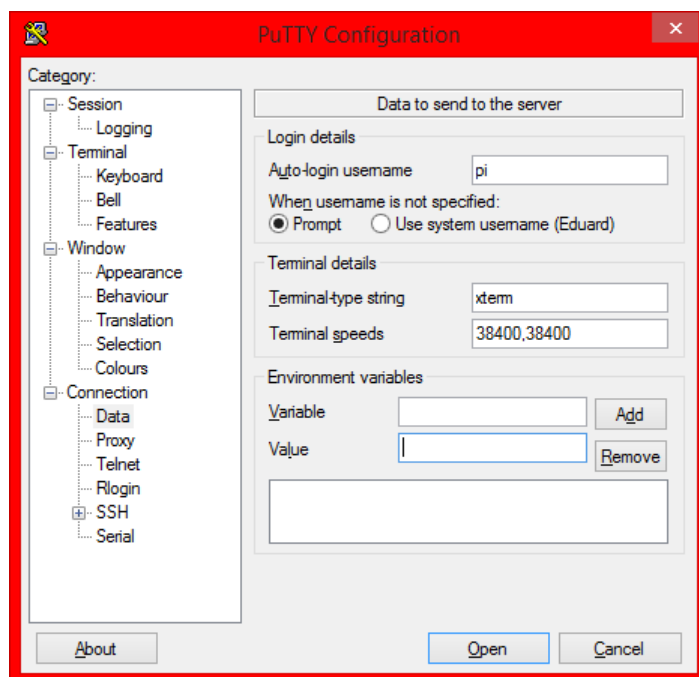


Hier gibt man zuerst die IP-Adresse des Raspberry Pi ein. Um die IP-Adresse des Raspberry Pi herauszufinden gibt man in dem Terminal „ip a“ ein.

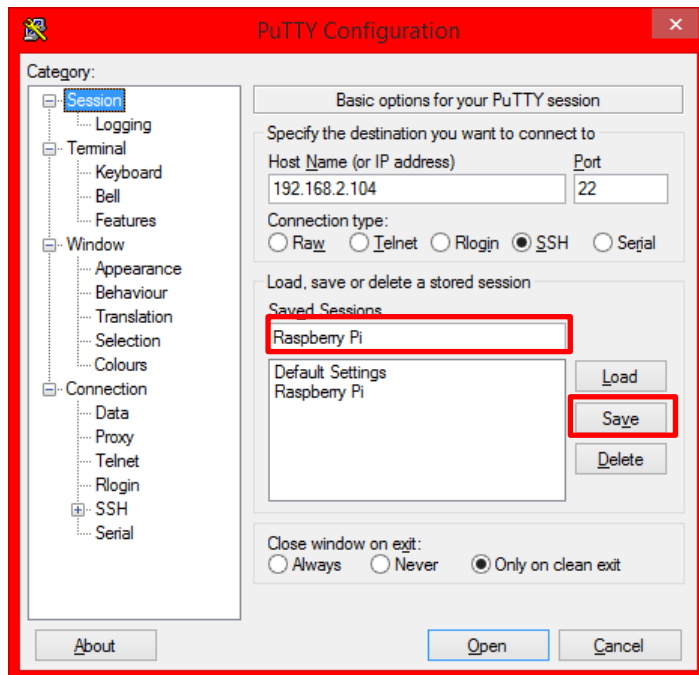
Als nächstes wählt man das Feld „Data“ aus.



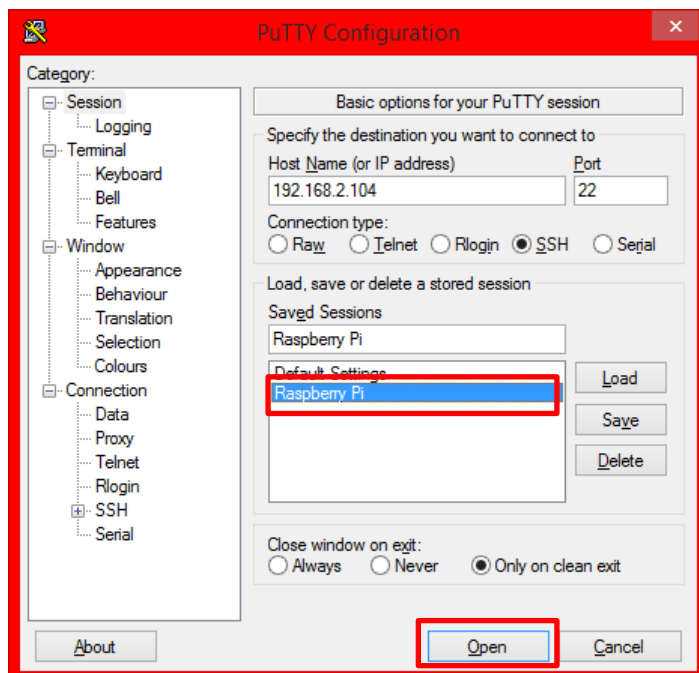
Hier gibt man in das „auto-login username“-Feld „pi“ ein.



Damit die Einstellungen nach jedem Aufruf von PuTTY nicht erneut eingegeben werden müssen, speichert man die Einstellungen. Dazu gibt man zuerst den Namen, unter welchem die Einstellungen gespeichert werden sollen und drückt dann auf den „Save“-Button.



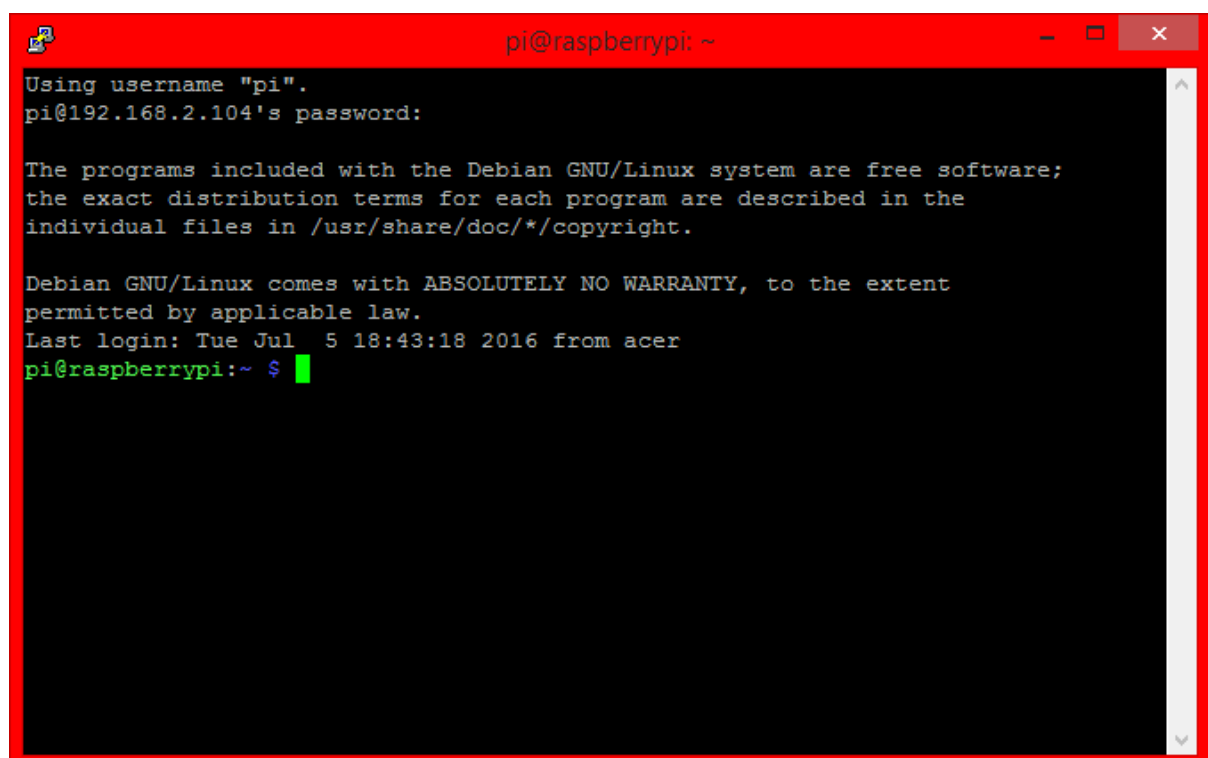
Schließlich wählt man die Session und drückt auf den „Open“-Button.





Als nächstes gibt man das Passwort ein. Wenn dieses bisher nicht geändert wurde, lautet es „raspberrry“.

Nach der Eingabe des Passwortes kann das Terminal des Raspberry Pi vom Computer aus bedient werden.

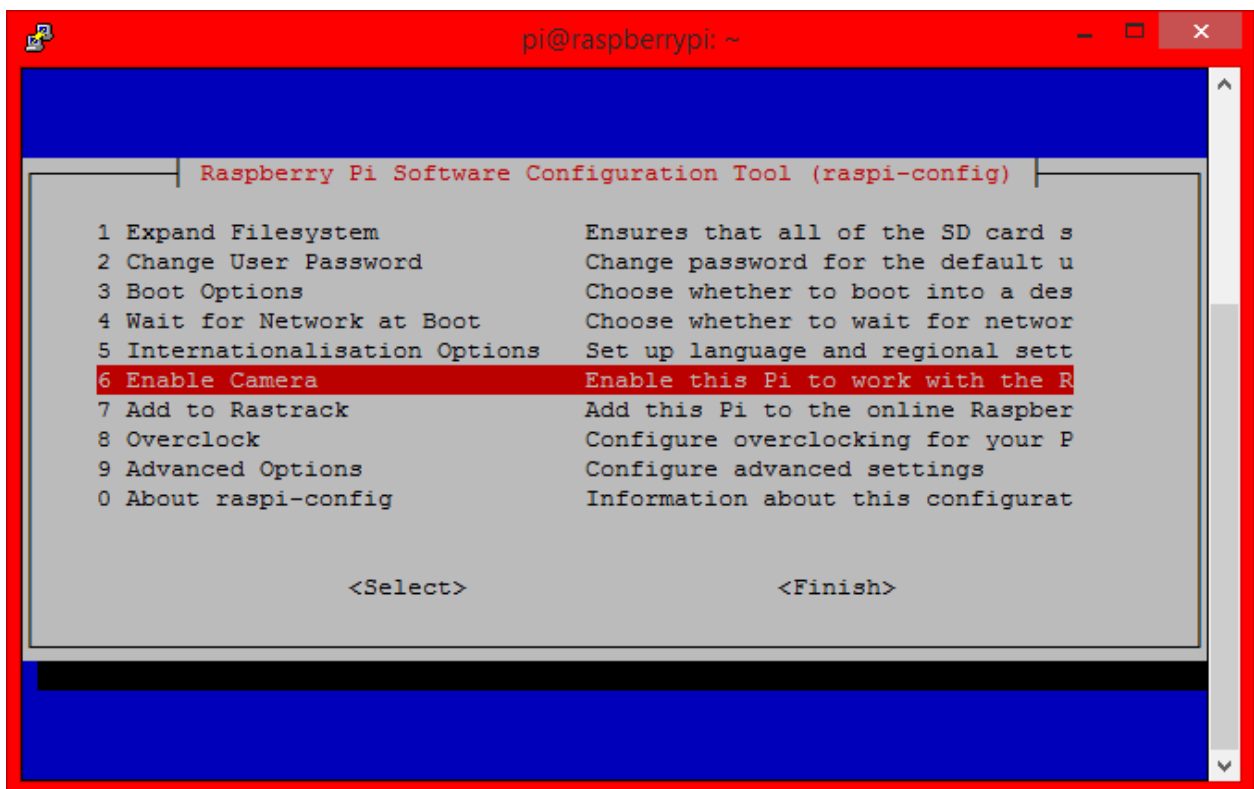


Siehe:

- <http://www.pc-magazin.de/bildergalerie/raspberry-pi-ssh-putty-windows-remote-control-access-zugriff-galerie-2525708-434300.html>
- <http://www.raspberrypi-tutorials.de/software/ssh-zugriff-auf-dem-raspberry-pi-einrichten.html>
- <http://www.forum-raspberrypi.de/Thread-tutorial-ssh-zugang-auf-dem-raspberry-pi-einrichten>

Raspberry Pi-Kamera aktivieren

Da das Endsystem eine Kamera zum Erstellen von Fotos verwendet, muss diese wie die SSH-Verbindung aktiviert werden. Dazu wählt man wieder das Terminal aus und gibt „sudo raspi-config“ ein. In dem „Configuration Tool“ wählt man „Enable Camera“ aus und setzt diese auf „Enable“. Als nächstes führt wird ein Neustart des Systems durchgeführt. Dadurch ist die Kamera des Raspberry Pi aktiviert und kann verwendet werden.

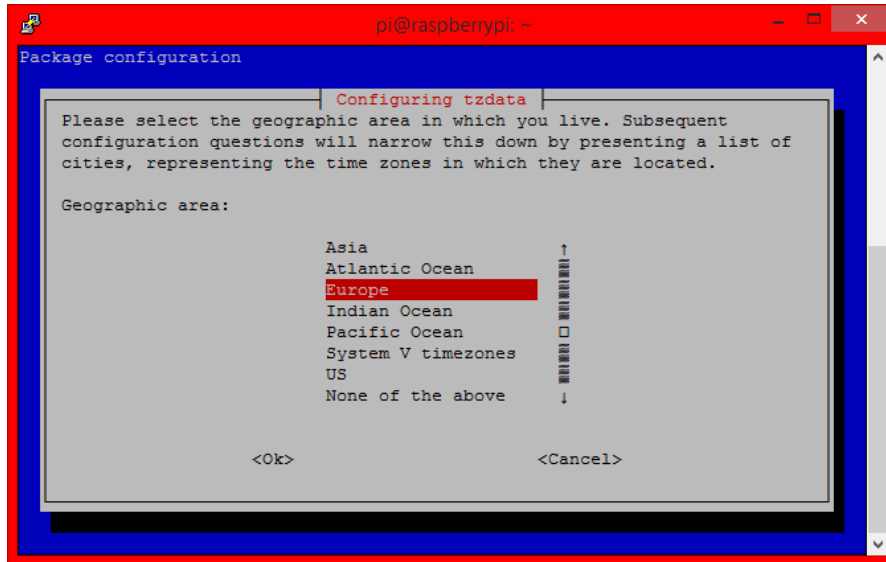


Siehe:

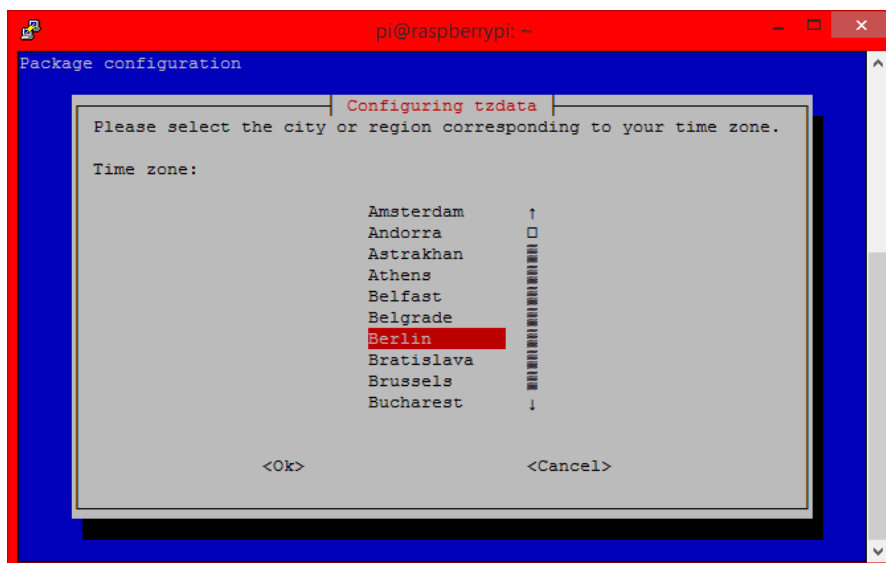
- http://www.netzmafia.de/skripten/hardware/RasPi/RasPi_Kamera.html

Raspberry Pi Zeitzone einstellen

Da das Endsystem eine Kamera zum Erstellen von Fotos verwendet, und diese nur zur bestimmten Zeiten aktiviert werden soll, muss die Zeitzone auf dem Raspberry Pi eingestellt werden. In der JaRasPi-Anwendung wird die Europäische Zeitzone verwendet, deshalb muss diese im Raspberry Pi eingestellt werden. Dazu wird das Raspberry Pi Terminal aufgerufen und folgender Befehl eingegeben: „sudo dpkg-reconfigure tzdata“. Es erscheint ein Fenster in dem die Zeitzone ausgewählt werden kann. Hier wird „Europe“ ausgewählt.



Als nächstes muss die Stadt ausgewählt werden. Hier wird „Berlin“ ausgewählt.



Nachdem Berlin ausgewählt wurde, ist die Zeitzone eingestellt und die JaRasPi-Anwendung wird richtig ausgeführt.

Siehe:

- <https://jankarres.de/2015/04/raspberry-pi-zeitzone-aendern/>
- <http://hanspeter314.blogspot.de/2013/03/raspberry-pi-zeitzone-andern.html>

Java installieren

Das zu erstellende System arbeitet unter anderem mit Java. Dieses muss auf dem Raspberry Pi installiert werden. Bevor Java installiert werden kann, muss das OS auf den neusten Stand gebracht werden. Dies wird durch die Eingabe des Befehls „sudo apt-get update“ in dem Terminal erreicht. Als nächstes wird folgender Befehl eingegeben „sudo apt-get install openjdk-8-jre“ und ausgeführt. Dabei wird gefragt, ob die Installation fortgesetzt werden sollte. Hier wird Ja beziehungsweise „Y/y“ ausgewählt. Dadurch wird Java auf dem Raspberry Pi installiert.

Siehe:

- <http://www.forum-raspberrypi.de/Thread-tutorial-java-installation-von-java>

MySQL installieren

Da das zu erstellende System mit einer MySQL-Datenbank arbeitet, muss diese installiert werden. Dazu wird das Terminal aufgerufen und folgender Befehl eingegeben: „sudo apt-get install mysql-server mysql-client php5-mysql“. Dabei wird gefragt, ob die Installation fortgesetzt werden sollte. Hier wird Ja beziehungsweise „Y/y“ ausgewählt. Dadurch wird MySQL auf dem Raspberry Pi installiert.

Wichtig: Während der Installation muss ein Passwort für die MySQL-Datenbank erstellt werden. Dieses Passwort ist wichtig, um mit der Datenbank arbeiten zu können.

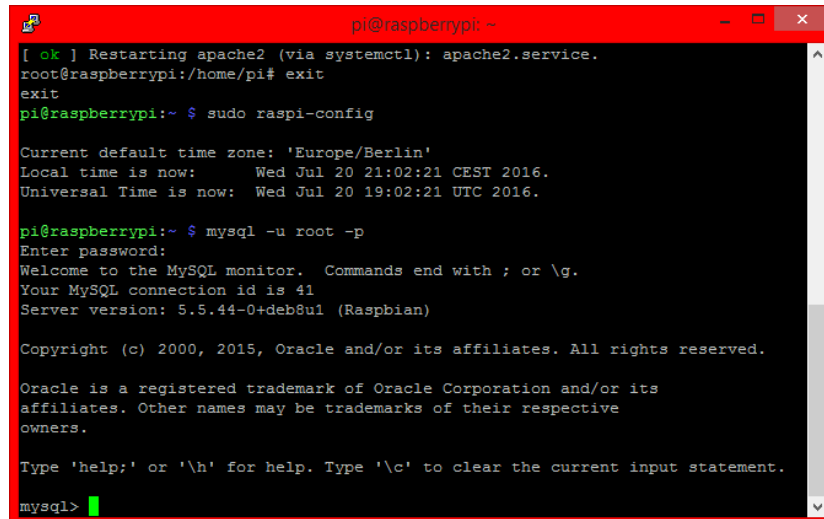
Siehe:

- http://kampus-elektroecke.de/?page_id=2717
- <http://www.manfred-steger.de/de/tuts/20-der-eigene-webserver-mit-dem-raspberry-pi#>

MySQL Datenbankeinträge

Da für die JaRasPi-Anwendung die MySQL-Datenbank verwendet wird, müssen folgende Befehle ausgeführt werden.

Zuerst muss nachdem MySQL installiert wurde, folgender Befehl eingegeben werden „sudo mysql -u root -p“. Es muss das zuvor definierte Passwort eingegeben werden. Wurde alles richtig eingegeben, wird in der Console folgendes angezeigt:



```

pi@raspberrypi: ~
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@raspberrypi:/home/pi# exit
exit
pi@raspberrypi:~ $ sudo raspi-config

Current default time zone: 'Europe/Berlin'
Local time is now:        Wed Jul 20 21:02:21 CEST 2016.
Universal Time is now:    Wed Jul 20 19:02:21 UTC 2016.

pi@raspberrypi:~ $ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 5.5.44-0+deb8u1 (Raspbian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

Jetzt können die MySQL-Befehle ausgeführt werden.

Siehe:

- http://kampus-elektroecke.de/?page_id=2717

Mit dem Befehl „mysql -u user -p < Datenbank.sql“ kann die benötigte JaRasPi-Datenbank auf einmal aufgesetzt werden. Die SQL-Datei „Datenbank.sql“ ist auf der CD vorhanden und kann hierzu verwendet werden.

Siehe:

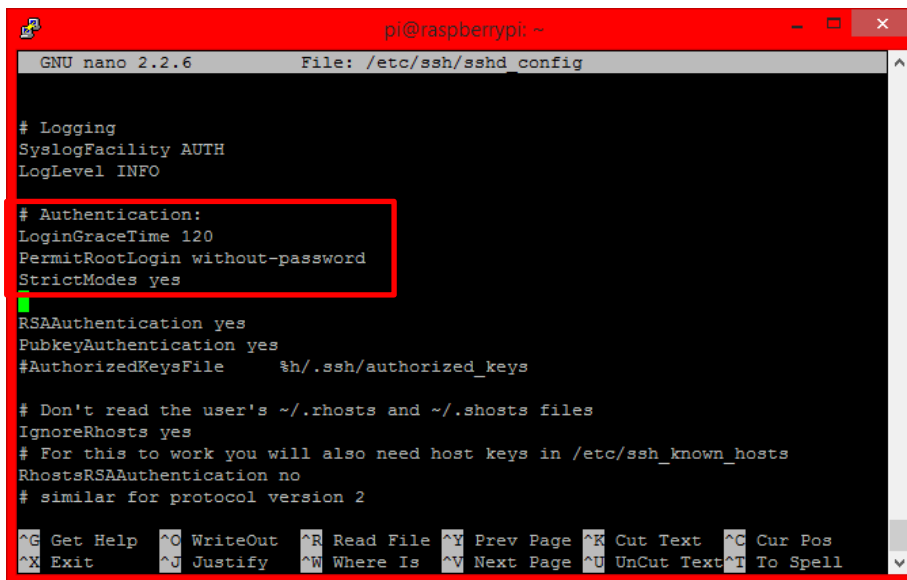
- <http://stackoverflow.com/questions/7616520/how-do-execute-a-sql-script-from-bash>

Root-Rechte einstellen

Damit der Raspberry Pi als Root verwendet werden kann, müssen einige Einstellungen vorgenommen werden. Es muss die Root-Passwort-Abfrage eingeschaltet werden. Folgende Befehle müssen in das Raspberry Pi Terminal eingegeben werden.

- `sudo su`
- `nano /etc/ssh/sshd_config`

Als nächstes wird nach folgendem Block gesucht:



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/ssh/sshd_config

# Logging
SyslogFacility AUTH
LogLevel INFO

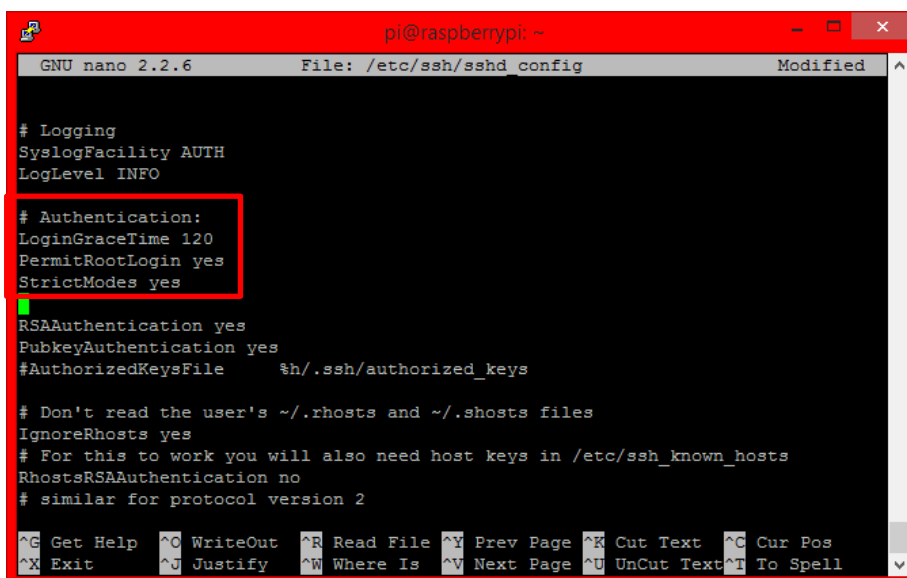
# Authentication:
LoginGraceTime 120
PermitRootLogin without-password
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Dieser Block muss wie folgt geändert werden:



```
pi@raspberrypi: ~
GNU nano 2.2.6 File: /etc/ssh/sshd_config Modified

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Mit Strg + x wird die Eingabe beendet. Als nächstes werden die Änderungen mit der Taste y/Y und Enter bestätigt.

Siehe:

- <http://www.forum-raspberrypi.de/Thread-der-pi-winscp-und-rootrechte>

Root-Passwort einstellen

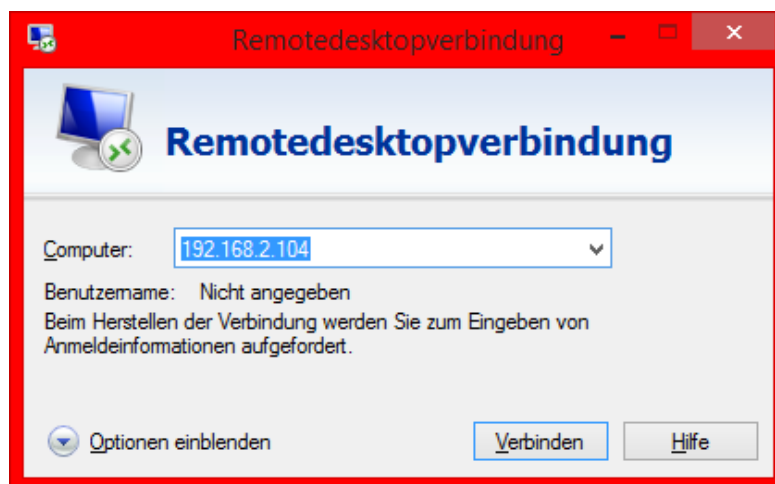
Soll die Remotedesktopverbindung als Root verwendet werden, muss das Root Passwort eingestellt werden. Dazu wird das Terminal aufgerufen und folgender Befehl eingegeben: „sudo passwd root“. Im Anschluss wird das Passwort zweimal eingegeben. Somit wurde das Passwort für den Root gesetzt.

Siehe:

- <http://www.forum-raspberrypi.de/Thread-der-pi-winscp-und-rootrechte>

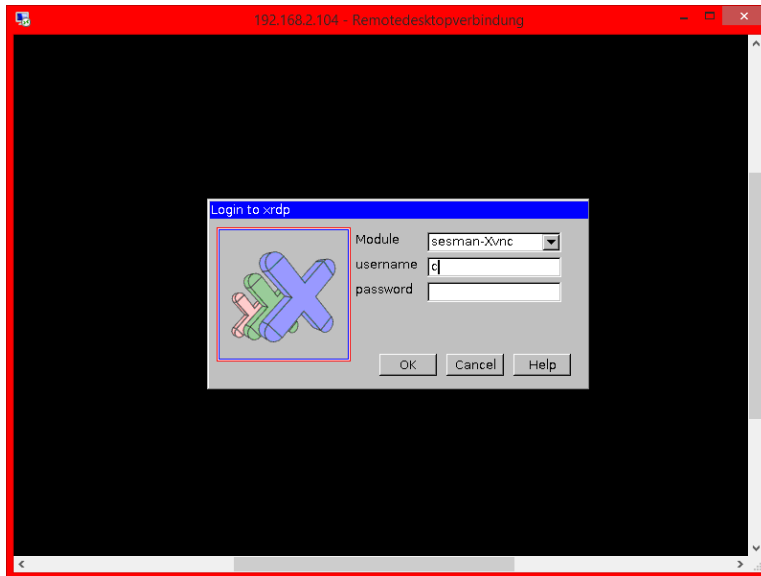
Remotedesktopverbindung

Neben der Verwendung des Raspberry Pi Terminals, ist die Verwendung des Desktops vorteilhaft. Um den Anzeigebildschirm des Raspberry Pi auf eigenem Computer zu verwenden, müssen folgende Punkte abgearbeitet werden. Zum einen wird das Programm „Remotedesktopverbindung“ benötigt. Wird das Programm ausgeführt erscheint folgendes Fenster:



Hier muss die IP-Adresse des Raspberry Pi eingegeben werden.

Bevor jedoch die Verbindung mit dem Raspberry Pi hergestellt werden kann, muss im Terminal des Micro-Computers folgender Befehl „sudo apt-get install xrdp“ ausgeführt und mit Y/y bestätigt werden. Nach der Installation kann die Remoteverbindung hergestellt werden.



Hier muss der Benutzername (pi) und das Passwort (raspberrypi) eingegeben werden. Nach dem Eingeben erscheint der Desktop des Raspberry Pi.

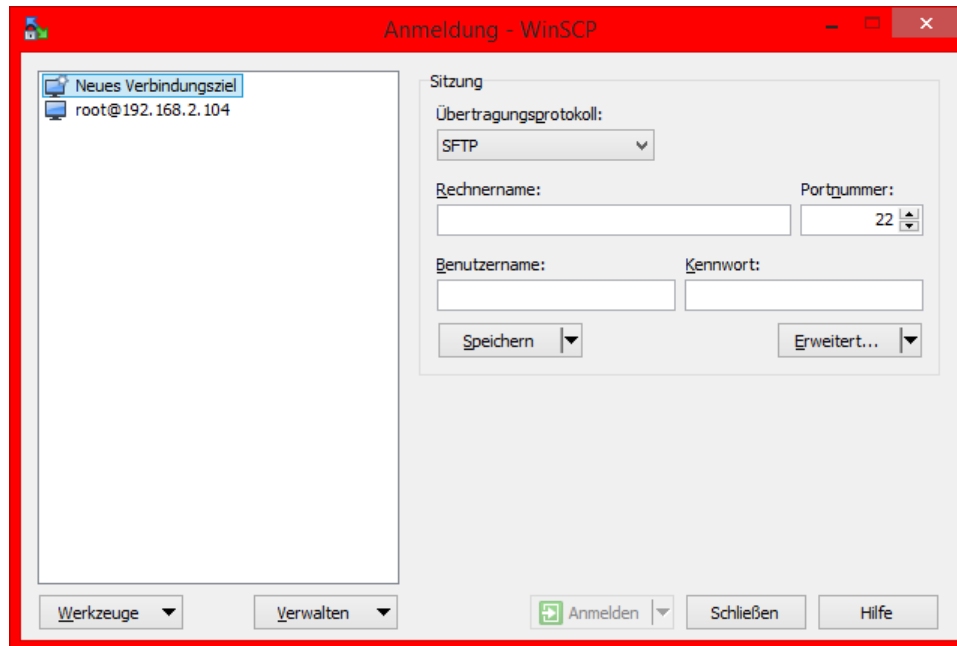
Möchte man sich als Root anmelden, muss der Benutzername (root) und das Passwort, welches man für Root angelegt hat, eingegeben werden.

Siehe;

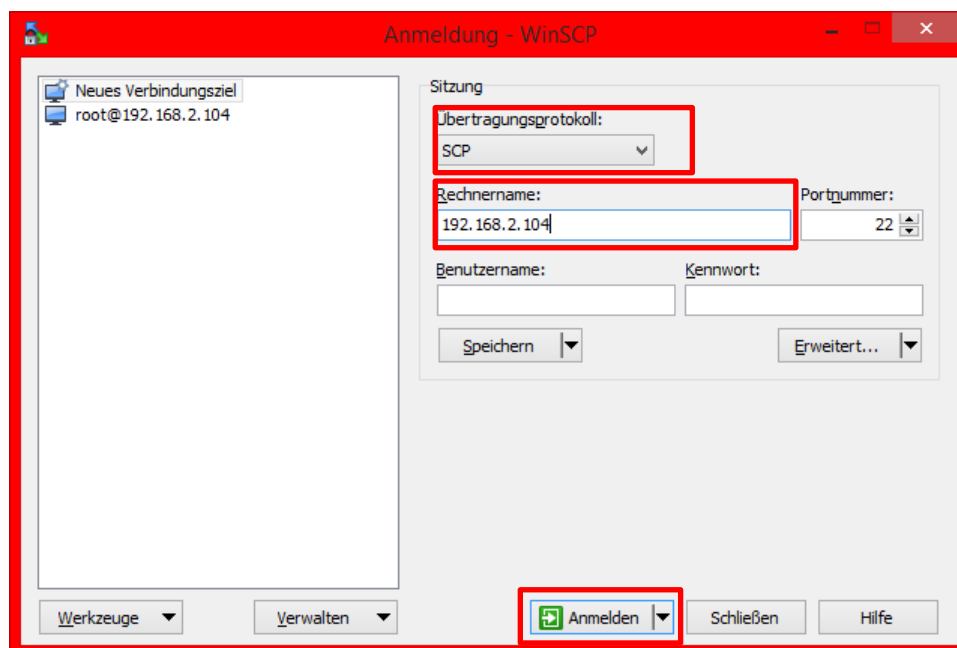
- <https://jankarres.de/2014/02/raspberry-pi-remote-desktop-installieren/>

WinSCP

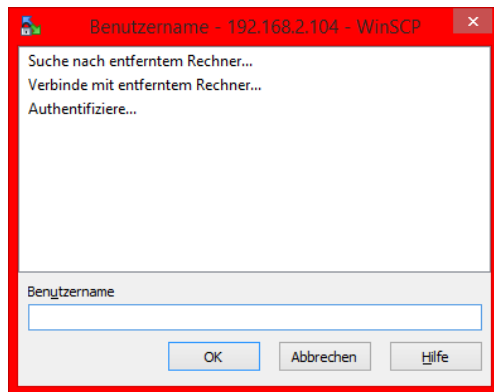
Mit dem kostenlosen Programm „WinSCP“ lassen sich Daten vom Computer auf den Raspberry Pi problemlos kopieren. Nachdem WinSCP installiert und geöffnet wurde erscheint folgendes Fenster:



Hier wird das Übertragungsprotokoll (SCP) ausgewählt und der Rechnername (IP-Adresse) eingegeben. Anschließend wird der Button „Anmelden“ gedrückt.



Als nächstes wird der Benutzername und das Passwort eingegeben.

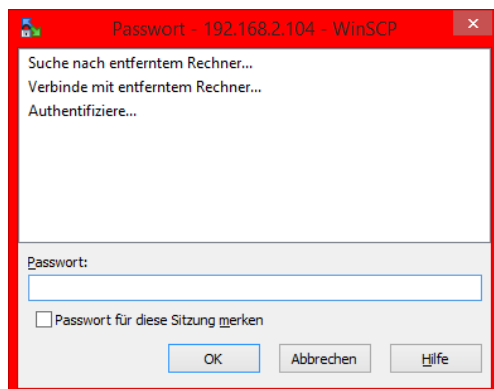


Benutzername - 192.168.2.104 - WinSCP

Suche nach entferntem Rechner...
Verbinde mit entferntem Rechner...
Authentifiziere...

Benutzername

OK Abbrechen Hilfe



Passwort - 192.168.2.104 - WinSCP

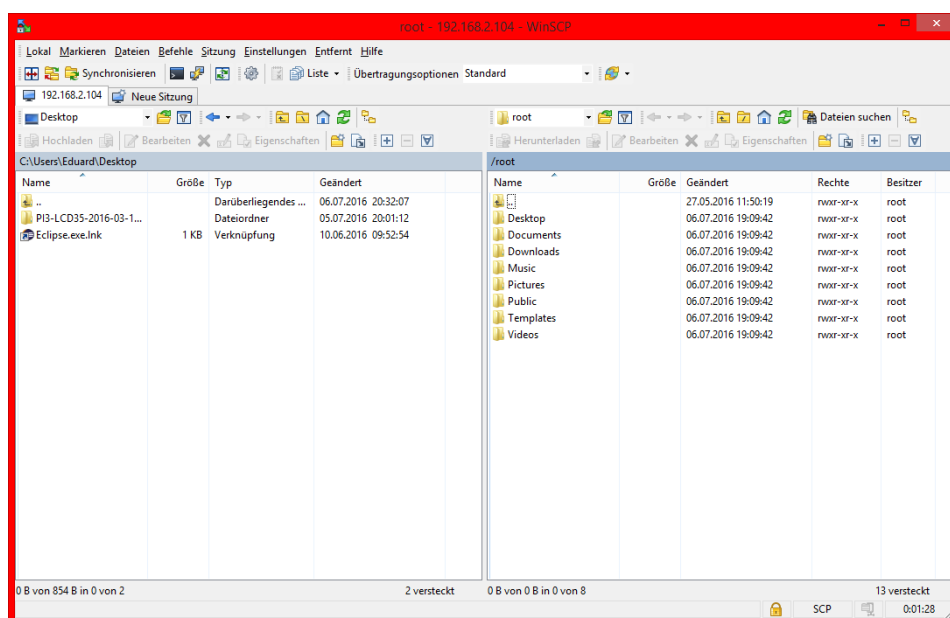
Suche nach entferntem Rechner...
Verbinde mit entferntem Rechner...
Authentifiziere...

Passwort:

☐ Passwort für diese Sitzung merken

OK Abbrechen Hilfe

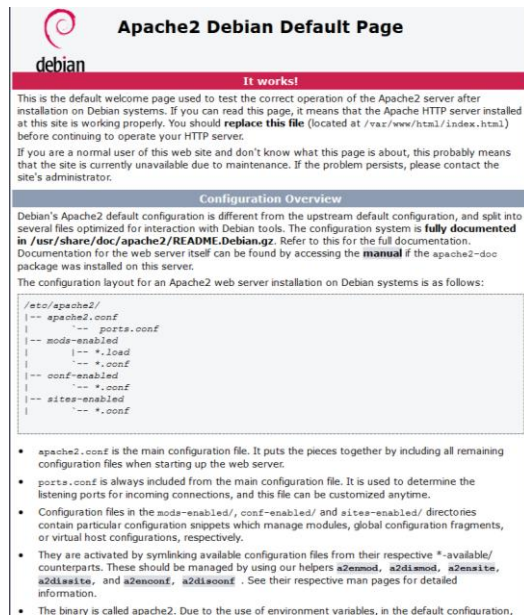
Nach der Eingabe erscheint das WinSCP-Fenster.



Apache 2 installieren

Möchte der Benutzer einen Webserver auf dem Raspberry Pi verwenden, muss der Apache 2-Webserver auf dem Gerät installiert werden. Mit dem Befehl „`sudo apt-get install apache2`“ wird der Webserver installiert. Die Installation muss mit y/Y bestätigt werden.

Um herauszufinden, ob die Installation erfolgreich gewesen ist, muss in den Browser, die IP-Adresse des Raspberry Pi eingegeben werden. Wurde der Apache 2 Server installiert, erscheint folgende Website:



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented** in `/usr/share/doc/apache2/README.Debian.gz`. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration.

Siehe:

- <http://www.forum-raspberrypi.de/Thread-tutorial-raspberry-pi-als-webserver-apache-2-installation>
- <https://www.raspberrypi.org/documentation/remote-access/web-server/apache.md>
- <http://www.manfred-steger.de/de/tuts/20-der-eigene-webserver-mit-dem-raspberry-pi#>

PHP installieren

Möchte der Benutzer auf dem Webserver PHP-Webseiten ausführen, muss PHP auf dem Raspberry Pi installiert werden. Mit dem Befehl „sudo apt-get install php5“ und dem Bestätigen, wird PHP auf dem Gerät installiert.

Um zu testen, ob PHP installiert wurde, wird in die Console „cd /var/www/html“ eingegeben. Als nächstes wird eine PHP-Datei durch folgenden Befehl „sudo nano phpinfo.php“ erstellt.





The screenshot shows a terminal window titled "pi@raspberrypi: /var/www". Inside, the GNU nano 2.2.6 text editor is open, editing a file named "phpinfo.php". The editor's content area shows the PHP opening and closing tags, followed by the phpinfo() function call: `<?php
phpinfo();
?>`. A red rectangular box highlights this code. The bottom of the window displays the standard nano editor command shortcuts, such as ^G for Get Help, ^O for WriteOut, and ^X for Exit.

Es erscheint folgendes Fenster, in dem folgender Befehl eingegeben wird „<?php phpinfo(); ?>“. Nach der Eingabe wird die Datei mit Strg + o gespeichert und die Eingabe mit Strg + x beendet.

Um zu testen, ob PHP installiert und die Datei erstellt wurde, wird in dem Webbrowser die IP-Adresse des Raspberry Pi und der Name der erstellten PHP-Datei, beispielsweise: „http://192.168.2.101/phpinfo.php“ eingegeben.

Das Ergebnis sollte wie folgt aussehen:

<div> <div>PHP Version 5.6.23-0+deb8u1</div>  </div>	
System	Linux raspberrypi 4.4.11-v7+ #888 SMP Mon May 23 20:10:33 BST 2016 armv7l
Build Date	Jul 14 2016 23:11:59
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysqli.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226.NTS
PHP Extension Build	API20131226.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk
<div> <div> This program makes use of the Zend Scripting Language Engine: Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies </div>  </div>	

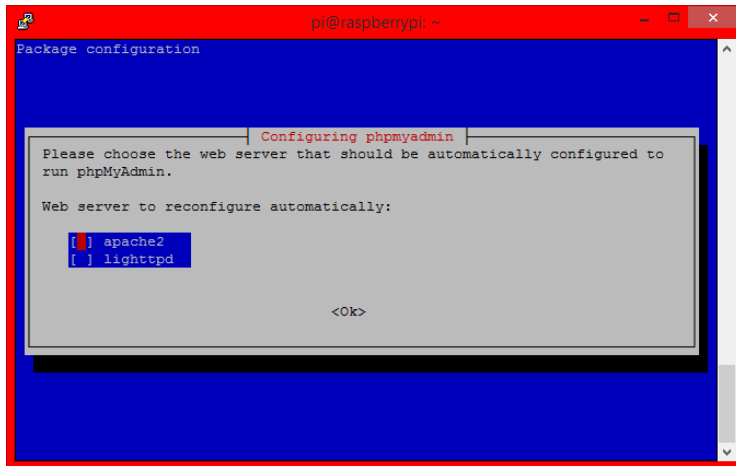
Configuration

Siehe:

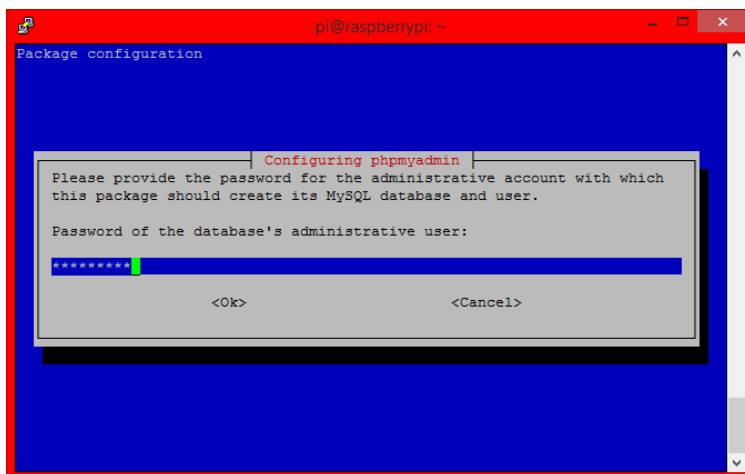
- <http://www.forum-raspberrypi.de/Thread-tutorial-raspberry-pi-als-webserver-php-5-installation>
- <http://www.manfred-steger.de/de/tuts/20-der-eigene-webserver-mit-dem-raspberry-pi#>

PHP-MyAdmin installieren

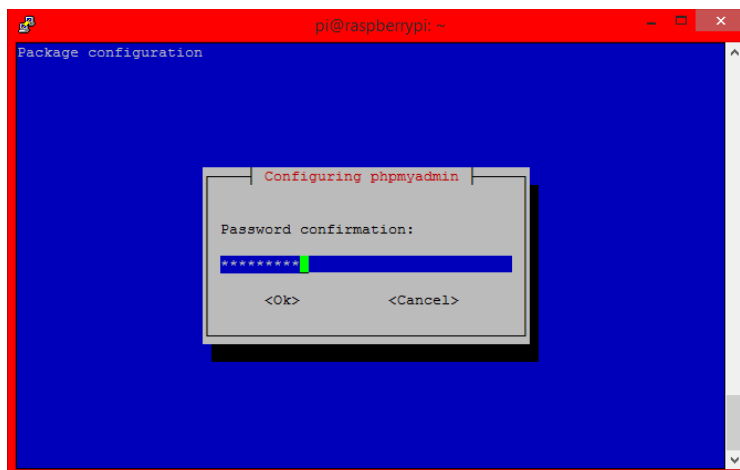
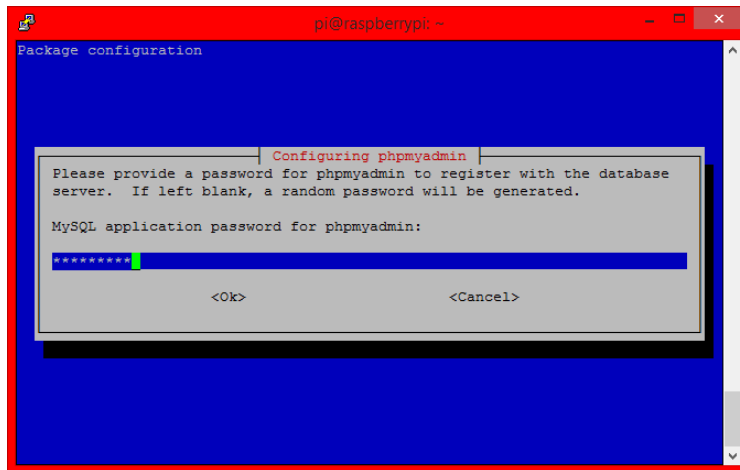
Um PHP-MyAdmin auf dem Raspberry Pi zu installieren müssen folgende Befehle ausgeführt werden. Für die Installation müssen Root-Rechte eingeholt werden, dies geschieht mit dem Befehl „sudo bash“. Als nächstes wird „apt-get install phpmyadmin“ eingegeben und mit y/Y bestätigt. Es erscheint ein Fenster in dem „apache 2“ ausgewählt werden muss.



Das nächste Fenster wird mit „Yes“ beantwortet. Es erscheint ein weiteres Fenster. Hier muss das zuvor erstellte MySQL Passwort eingegeben werden.



Wurde das Passwort eingegeben, wird ein Passwort für PHPMYAdmin festgelegt und bestätigt.



Wurden alle Daten richtig eingegeben wird PHPMyAdmin installiert. Als nächstes wird PHPMyAdmin unter Apache zum Ausführen gebracht. Dazu wird folgender Befehl „nano /etc/apache2/apache2.conf“ eingegeben. Es wird das File apache2.conf aufgerufen. Durch das Drücken von Strg + v kann in dem File weitergeblättert werden. Es muss bis zum Ende des Files weitergeblättert werden. Anschließend wird folgender Befehl „Include /etc/phpmyadmin/apache.conf“ eingegeben. Nach der Eingabe wird mit Strg + x das File geschlossen, davor aber mit Y gespeichert.

Als nächstes wird ein Neustart des Apache-Webserver mit folgendem Befehl „/etc/init.d/apache2 restart“ ausgeführt. Nun kann in dem Webserver PHPMyAdmin aufgerufen werden.



Willkommen bei phpMyAdmin

Sprache - Language

Deutsch - German

Anmeldung

Benutzername:

Passwort:

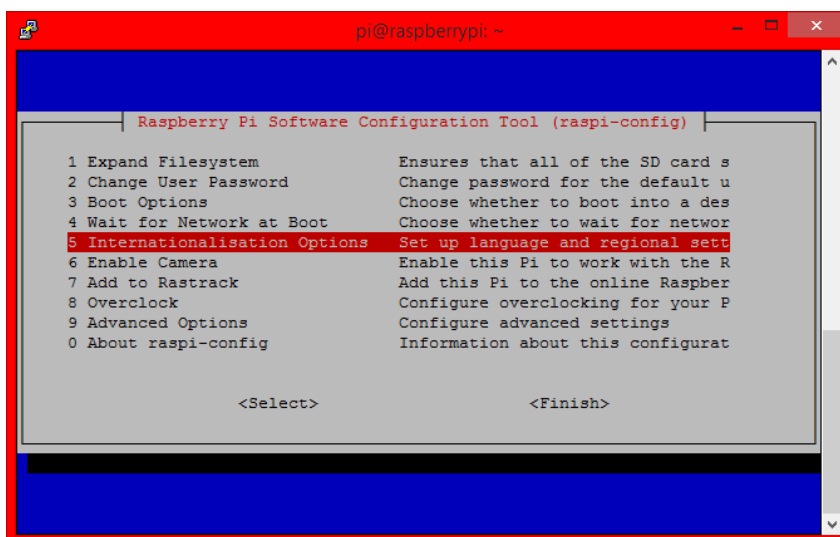
OK

Siehe:

- <http://www.forum-raspberrypi.de/Thread-tutorial-raspberry-pi-als-webserver-phpmyadmin-installation>
- <https://www.stewright.me/2012/09/tutorial-install-phpmyadmin-on-your-raspberry-pi/>
- <http://www.manfred-steger.de/de/tuts/20-der-eigene-webserver-mit-dem-raspberry-pi#>

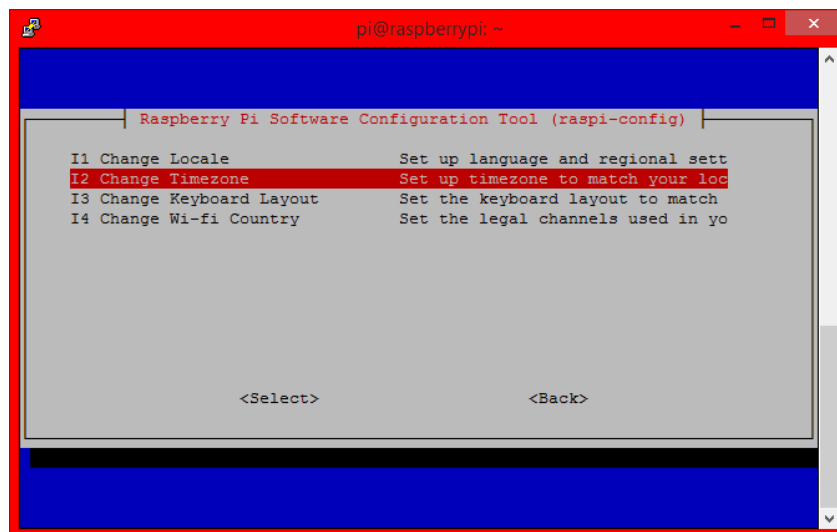
Raspberry Pi Zeitzone ändern

Dadurch, dass die JaRasPi-Anwendung zur bestimmten Uhrzeiten mit Hilfe der Kamera ein Foto erstellt und in der Anwendung die Zeit eingetragen wurde, muss die Zeitzone im Raspberry Pi eingetragen werden. Mit dem Befehl "sudo raspi-config" erscheint folgendes Fenster:



Hier wird „Internationalisation Options“ ausgewählt.

Danach wird „Change Timezone“ ausgewählt.



Anschließend muss zuerst "Europe" und dann "Berlin" ausgewählt werden.

Siehe:

- <https://jankarres.de/2015/04/raspberry-pi-zeitzone-aendern/>
- <http://strobelstefan.org/?p=2682>

